

# CUNY SPS DATA 621 - CTG5 - HW5

*Betsy Rosalen, Gabrielle Bartomeo, Jeremy O'Brien, Lidiia Tronina, Rose Koh*

*May 15th, 2019*

## Contents

<b>1</b>	<b>DATA EXPLORATION</b>	<b>2</b>
1.1	Summary Statistics . . . . .	3
1.1.1	Summary Statistics Graphs . . . . .	3
1.2	Linearity . . . . .	6
1.2.1	Log Transformed Data . . . . .	7
1.2.2	Square Root Transformed Predictors and Log Transformed Target . . . . .	8
1.3	Missing Data . . . . .	9
<b>2</b>	<b>DATA PREPARATION</b>	<b>11</b>
2.1	Missing Values . . . . .	11
2.2	Transformation / Feature Engineering . . . . .	11
<b>3</b>	<b>BUILD MODELS</b>	<b>12</b>
3.1	Poisson Regression Model . . . . .	12
3.1.1	Quasipoisson Model . . . . .	15
3.1.2	Hurdle and Zero-inflated Poisson Models . . . . .	17
3.2	Negative binomial regression models . . . . .	24
3.2.1	Negative binomial regression model 1 . . . . .	24
3.2.2	Negative binomial regression model 2 . . . . .	26
3.3	Linear regression models . . . . .	27
3.3.1	Linear regression model 1 . . . . .	27
3.3.2	Linear regression model 2 . . . . .	29
<b>4</b>	<b>SELECT MODELS</b>	<b>32</b>
4.1	Comparison of models . . . . .	32
4.2	Diagnostic plots . . . . .	33
4.3	Prediction . . . . .	34
<b>5</b>	<b>Appendix</b>	<b>36</b>

Table 1: Data Dictionary

VARIABLE	DEFINITION	TYPE
TARGET	Number of Cases Purchased	count response
AcidIndex	Method of testing total acidity by using a weighted avg	continuous numerical predictor
Alcohol	Alcohol Content	continuous numerical predictor
Chlorides	Chloride content of wine	continuous numerical predictor
CitricAcid	Citric Acid Content	continuous numerical predictor
Density	Density of Wine	continuous numerical predictor
FixedAcidity	Fixed Acidity of Wine	continuous numerical predictor
FreeSulfurDioxide	Sulfur Dioxide content of wine	continuous numerical predictor
LabelAppeal	Marketing Score indicating the appeal of label design	categorical predictor
ResidualSugar	Residual Sugar of wine	continuous numerical predictor
STARS	Wine rating by a team of experts. 4 = Excellent, 1 = Poor	categorical predictor
Sulphates	Sulfate content of wine	continuous numerical predictor
TotalSulfurDioxide	Total Sulfur Dioxide of Wine	continuous numerical predictor
VolatileAcidity	Volatile Acid content of wine	continuous numerical predictor
pH	pH of wine	continuous numerical predictor

## 1 DATA EXPLORATION

When dining in a restaurant, a sommelier can assist you in selecting a perfect wine, even if you do not know much about wine yourself. By asking about your taste preferences, they can recommend a wine that pairs well with your meal, while complementing your likes and dislikes. But what happens when you're a large wine manufacturer, wondering how to produce wines that will sell? If the wine manufacturer can predict the number of cases sold based on the characteristics of a wine, then that manufacturer will be able to adjust their wine offering to maximize sales.

Our data set contains information on approximately 12,000 commercially available wines. Most of the variables are related to the chemical properties of the wine. The response variable is the number of sample cases that were purchased by wine distribution companies after sampling the wine. These cases would be used to provide tasting samples to restaurants and wine stores around the United States and promote further sales.

Table 2: Summary statistics for numeric variables

	n	min	mean	median	max	sd
AcidIndex	12795	4.00	7.77	8.00	17.0	1.32
Alcohol	12142	-4.70	10.49	10.40	26.5	3.73
Density	12795	0.89	0.99	0.99	1.1	0.03
Sulphates	11585	-3.13	0.53	0.50	4.2	0.93
pH	12400	0.48	3.21	3.20	6.1	0.68
TotalSulfurDioxide	12113	-823.00	120.71	123.00	1057.0	231.91
FreeSulfurDioxide	12148	-555.00	30.85	30.00	623.0	148.71
Chlorides	12157	-1.17	0.05	0.05	1.4	0.32
ResidualSugar	12179	-127.80	5.42	3.90	141.2	33.75
CitricAcid	12795	-3.24	0.31	0.31	3.9	0.86
VolatileAcidity	12795	-2.79	0.32	0.28	3.7	0.78
FixedAcidity	12795	-18.10	7.08	6.90	34.4	6.32
TARGET	12795	0.00	3.03	3.00	8.0	1.93

Table 3: Summary statistics for categorical variables

	STARS	LabelAppeal
1 :3042	-2: 504	
2 :3570	-1:3136	
3 :2212	0 :5617	
4 : 612	1 :3048	
NA's:3359	2 : 490	

## 1.1 Summary Statistics

Continuous quantitative and categorical variables were summarized separately for the sake of clarity.

`STARS` and `LabelAppeal` can be described as ordinal categorical variables. However, because of the numerical coding, these variables were imported as if they were quantitative. Since they are ordinal consideration was given to treating them as numerical, but ultimately the decision was made to convert them to factors.

### 1.1.1 Summary Statistics Graphs

The histograms in Figure 1 shows the distribution of all 15 variables. The distribution is more peaked than a normal bell curve for most of our variables especially `Chlorides`, `CitricAcid`, `Density`, `FixedAcidity`, `FreeSulfurDioxide`, `pH`, `ResidualSugar`, `Sulphates`, `TotalSulfurDioxide` and `VolatileAcidity`. Most of those are also centered at or near zero. `AcidIndex` has a slight right skew.

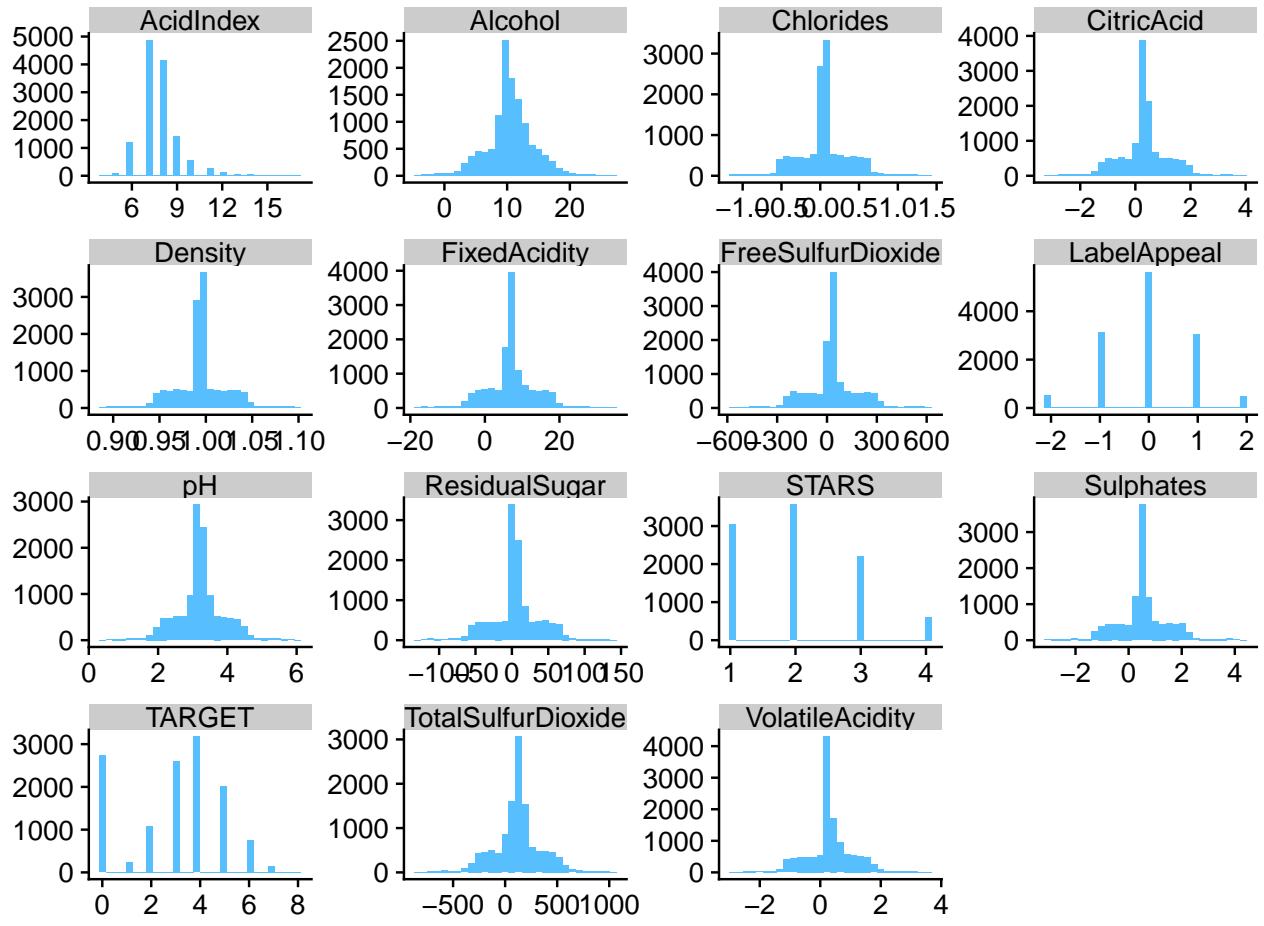


Figure 1: Data Distributions

We see in figures 2 and 3 that the target has very different distributions for the different levels of the other two categorical values. Both categorical variables, STARS and LabelAppeal, seem to have a positive linear relationship with the target. There is also some evidence of zero-inflation.

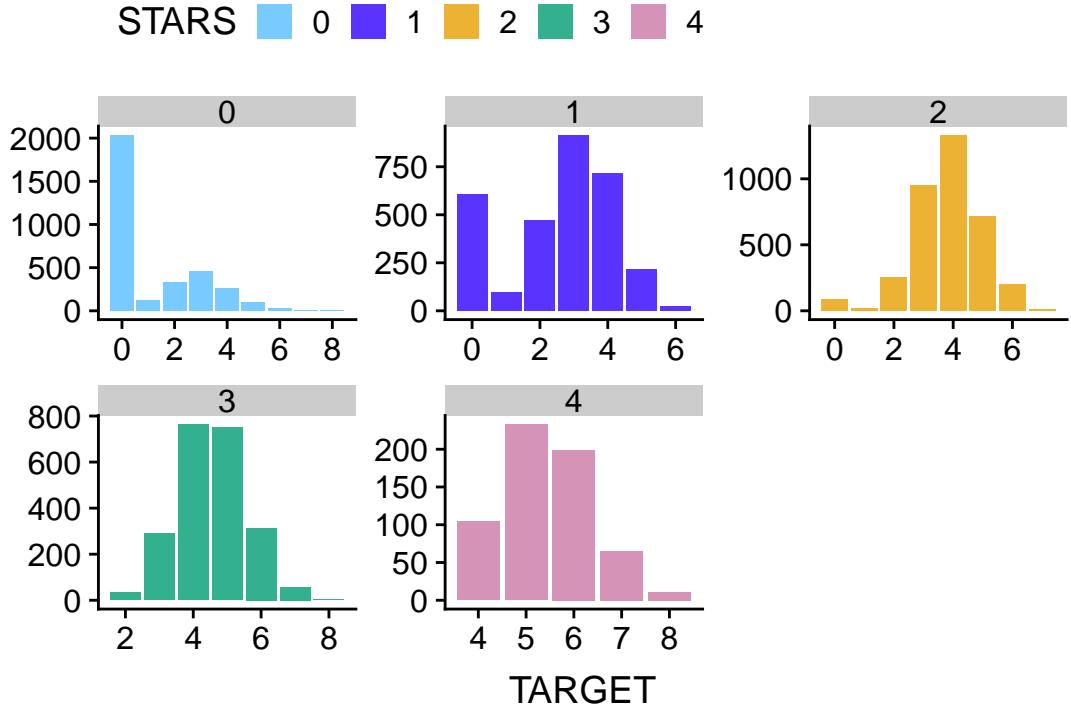


Figure 2: TARGET Distributions by STARS Values

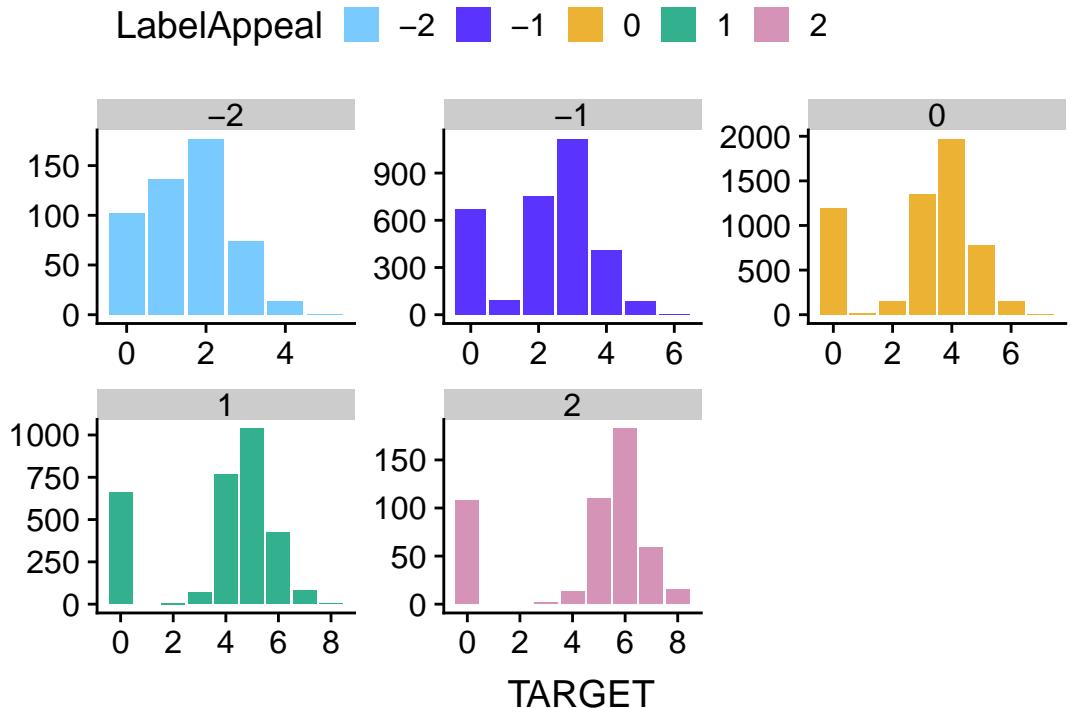


Figure 3: TARGET Distributions by LabelAppeal Values

Figure 4 shows that there are a large number of outliers that need to be accounted for, except for `LabelAppeal`, `AcidIndex` and `STARS` which have a limited number of variations.

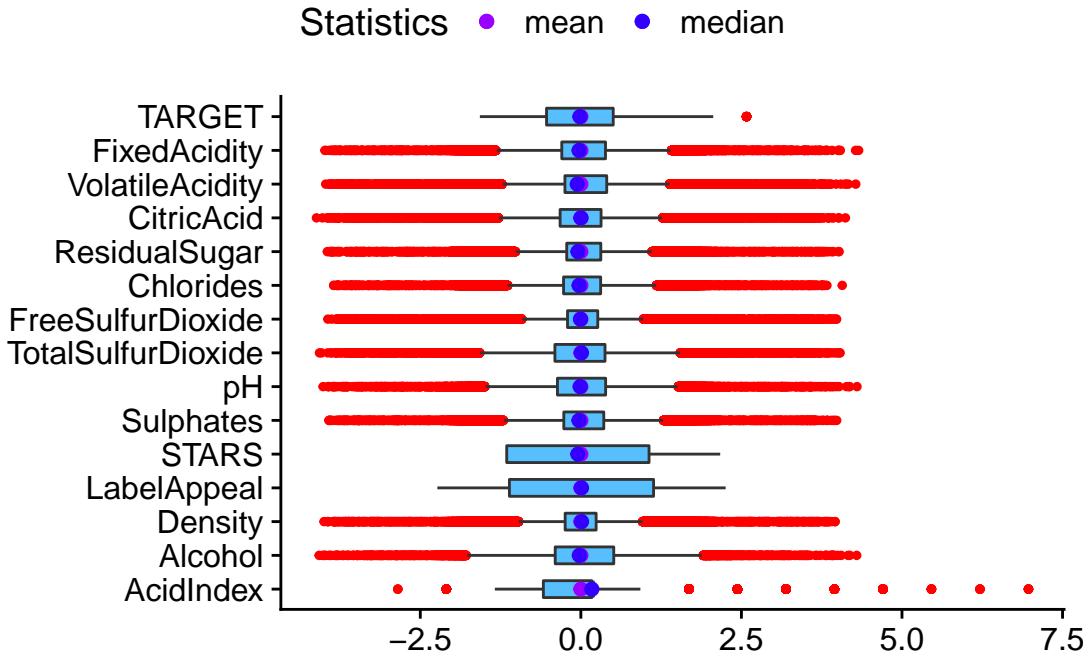


Figure 4: Scaled Boxplots

## 1.2 Linearity

The raw predictors fail to show a linear relationship with the TARGET except for the two categorical predictors, LabelAppeal and STARS, and AcidIndex. VolatileAcidity also shows a very slight negative linear relationship. The Scatter Plots show a systematic, wave-like pattern for Density, FixedAcidity, FreeSulfurDioxide, TotalSulfurDioxide and VolatileAcidity.

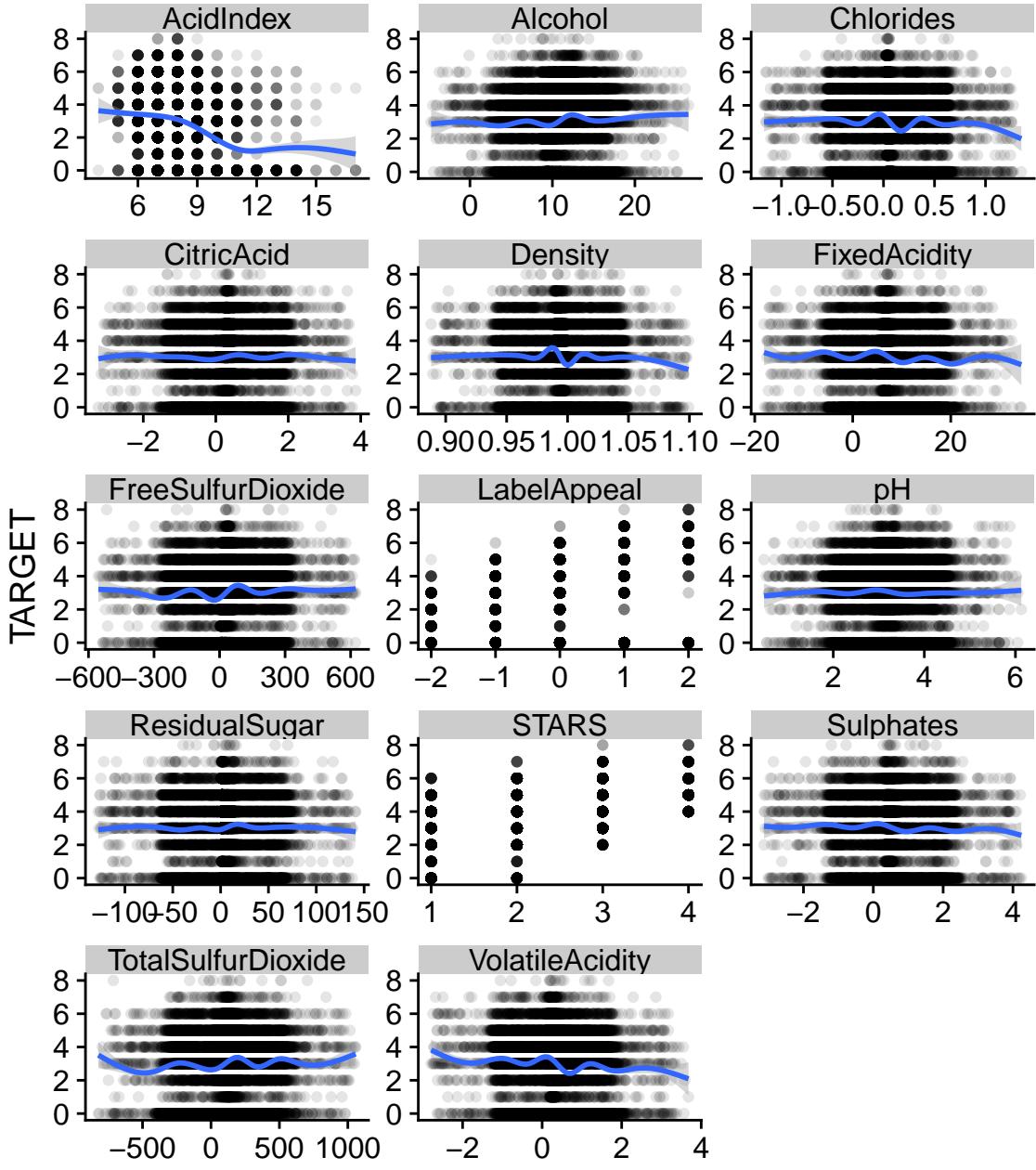


Figure 5: Scatter plot between numeric predictors and the TARGET

### 1.2.1 Log Transformed Data

In attempt to improve the linearity of the variables against the TARGET variable, we start with a log transformation on all predictors and TARGET variable. As a result, the linearity of Chlorides and FreeSulfurDioxide become more apparent.

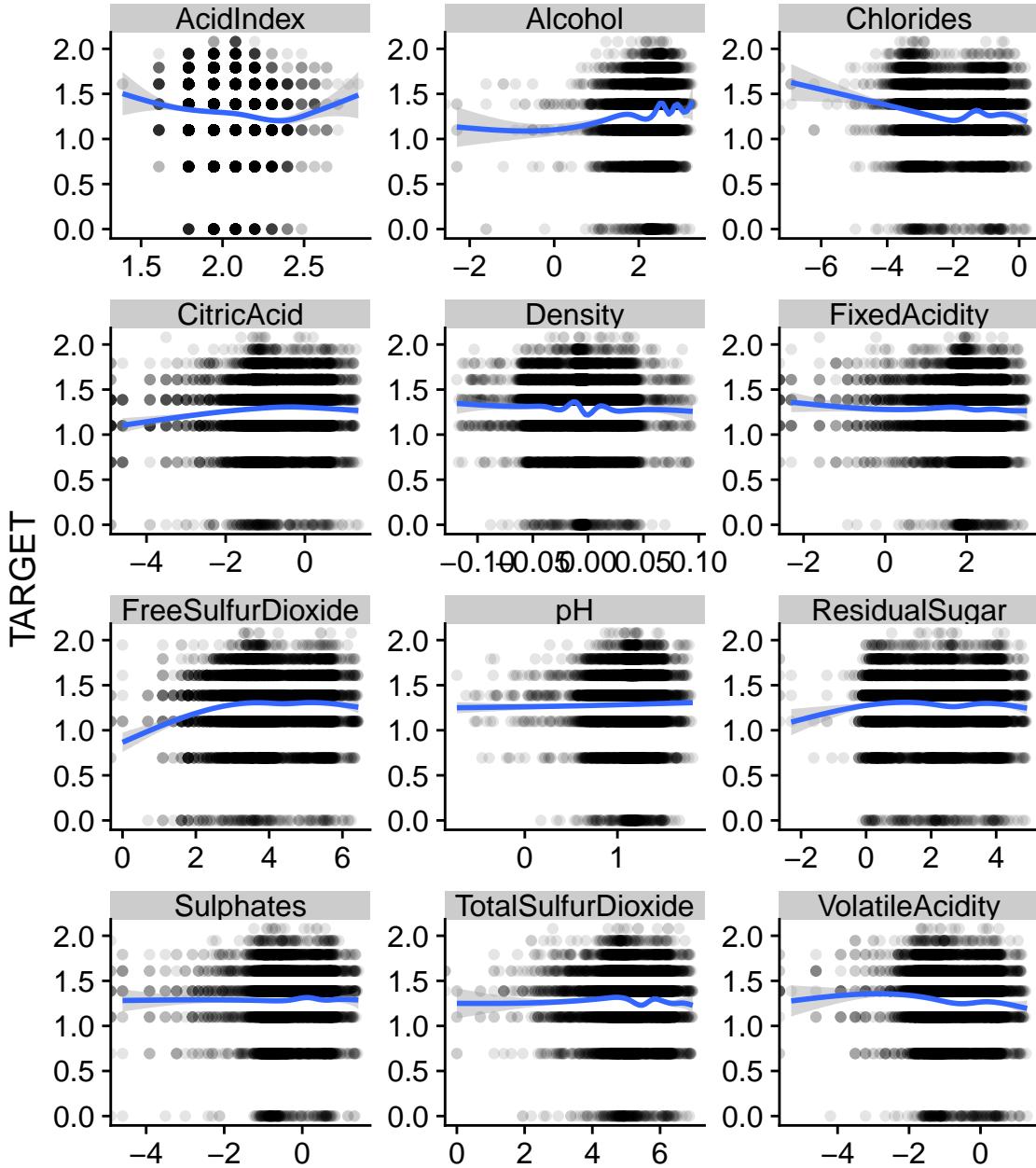


Figure 6: Scatter plot between log transformed predictors and the log transformed TARGET filtered for rows where TARGET is greater than 0

### 1.2.2 Square Root Transformed Predictors and Log Transformed Target

In ‘Linear Models with R’, Faraway suggested that the square root transformation is often appropriate for count response data. The Poisson distribution is a good model for counts, and that distribution has the property that the mean is equal to the variance thus suggesting the square root transformation. A plot of each predictor square root transformed plotted against the log transformed TARGET.

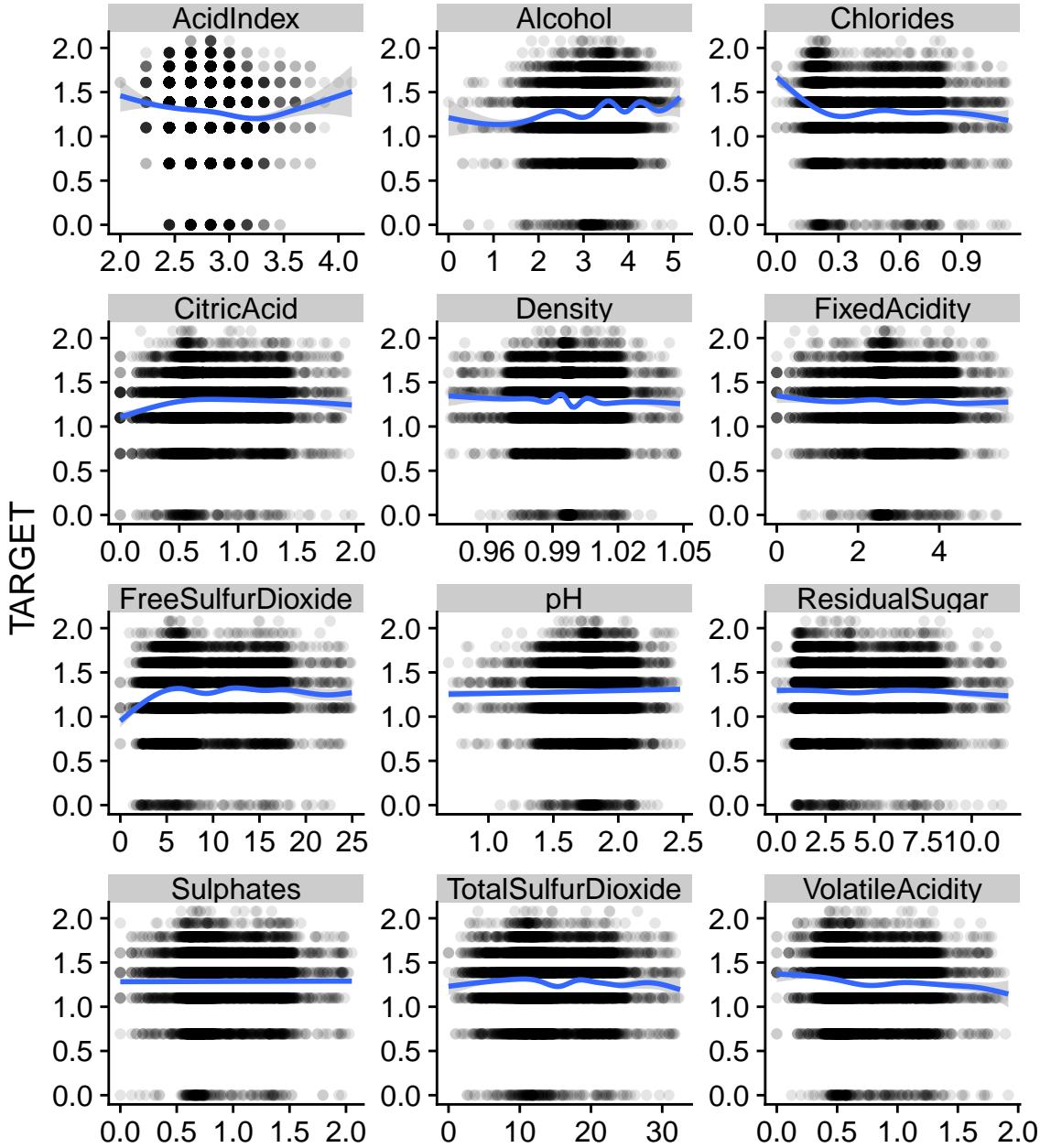


Figure 7: Scatter plot between square root transformed predictors and the square root transformed TARGET filtered for rows where TARGET is greater than 0

### 1.3 Missing Data

A number of variables are missing observations: STARS, Sulphates, TotalSulfurDioxide, Alcohol, FreeSulfurDioxide, Chlorides, ResidualSugar, pH. For STARS, the number is 26.25%, but the others range between 3% and 9% of total. Approximately 50% of the cases are missing one of these variables.

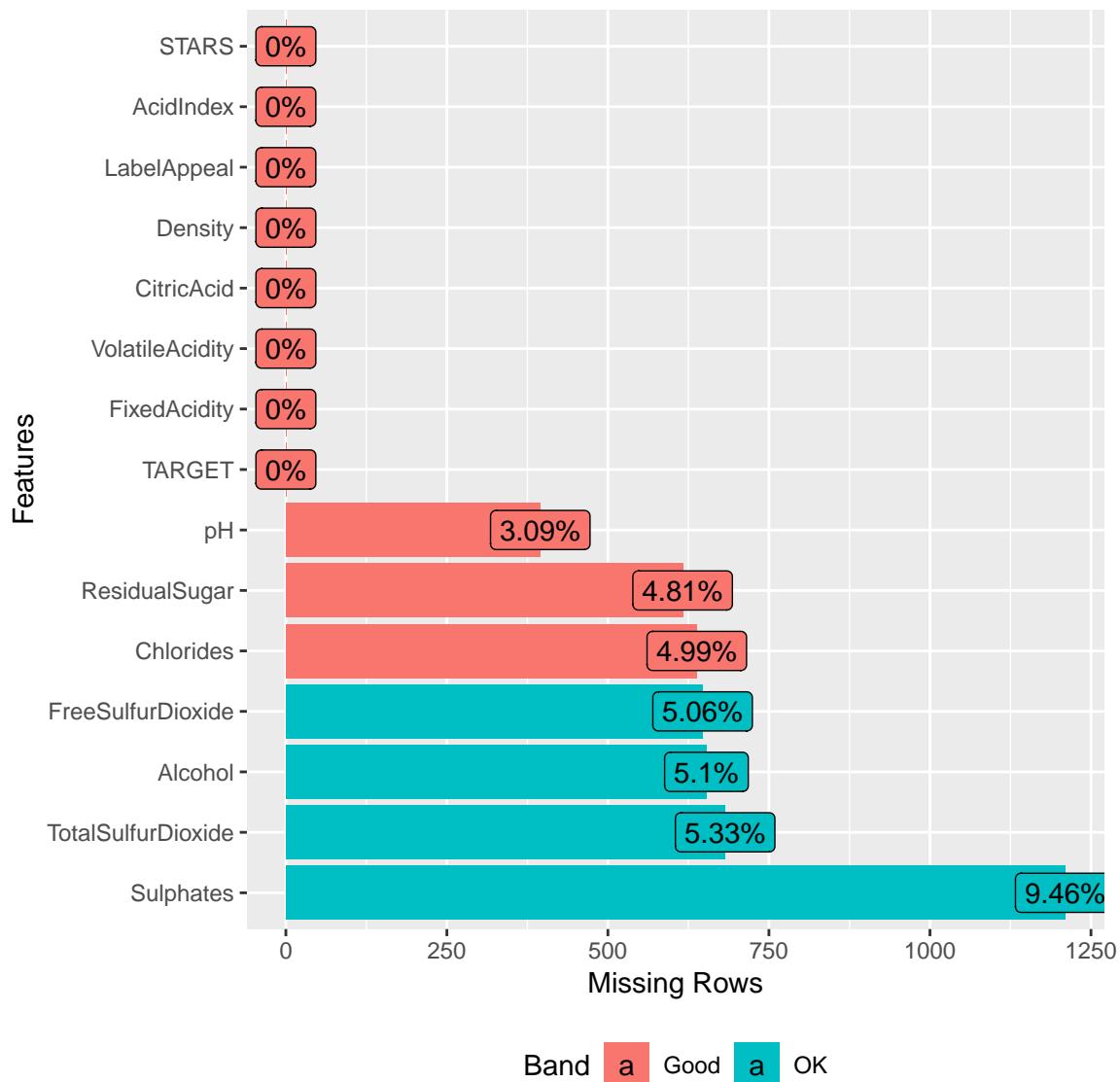


Figure 8: Missing data

## 2 DATA PREPARATION

### 2.1 Missing Values

There are many observations that are missing multiple pieces of data. Especially the STARS miss more than a quarter in both train and test dataset. This does not appear to be missing random pattern thus imputation would not be suitable. Moreover, it appears to be a highly significant predictor. To manage this, we simply assign 0 to NA values in STARS. The following variables are imputed using MICE package: Sulphates, TotalSulfurDioxide, Alcohol, FreeSulfurDioxide, Chlorides, ResidualSugar, or pH.

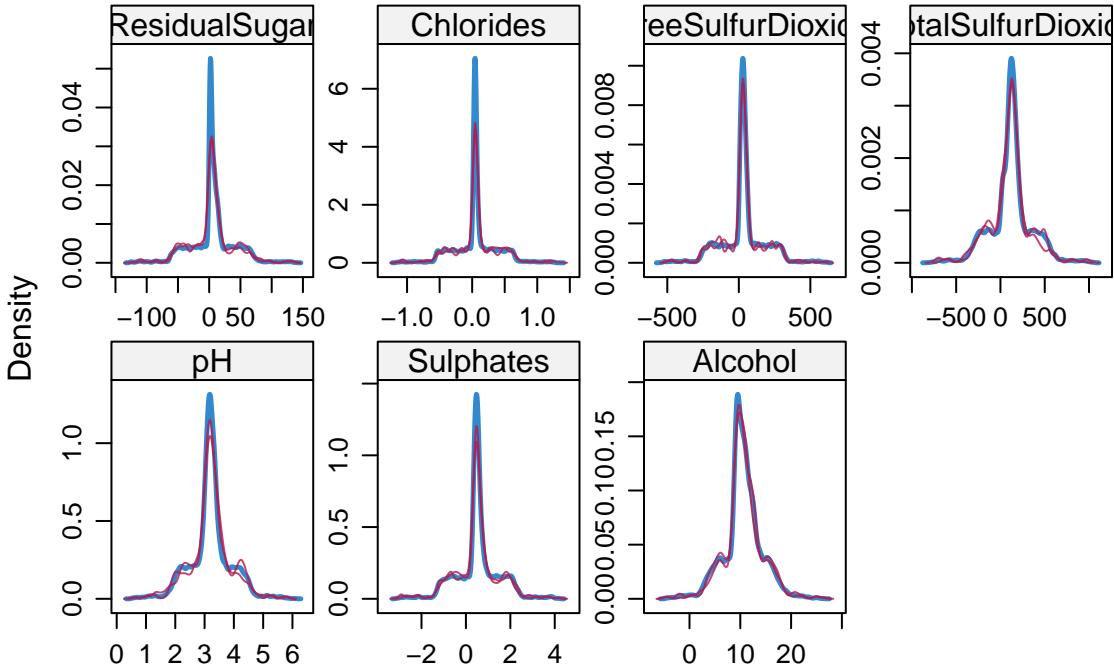


Figure 9: Difference between original and imputed data

Pink and blue lines indicate close fit match in distribution of imputed values and recorded values, with the exception of STARS - the addition of STARS values has given the appearance of shifting the distribution from high to low lambda values.

### 2.2 Transformation / Feature Engineering

There are a large number of negative values for variables for which negative values do not make logical sense, examples: Alcohol, CitricAcid, FixedAcidity, FreeSulfurDioxide, ResidualSugar, Sulphates, TotalSulfurDioxide, and VolatileAcidity. The range for the Poisson and negative binomial distribution has zero as a lower bound, so we can arithmetically transform the aforementioned variables to scale the lower IQR non-outlier values from zero up and drop the sub-IQR values (now the only negative values remaining).

Alternatively, we can explore whether more information on how measurements were made can be found to discern whether there might be some reason for negative values, and if there's a possibility of systematic data errors

We'll also test the data for over-dispersion when setting up negative binomial models.

Table 4: Comparison of different data prep methods influence on AIC

	df	AIC
imputed only	21	45620
plus min	21	45620
plus iqr1.5	21	45609
abs log	21	45636

Table 5: Comparison of different data prep methods influence on model coefficients

	imputed only	plus min	plus iqr1.5	abs log	range
(Intercept)	0.69	0.76	0.70	1.44	0.75
FixedAcidity	0.00	0.00	-0.27	-0.57	0.57
VolatileAcidity	-0.03	-0.03	-0.01	-0.04	0.03
CitricAcid	0.01	0.01	0.00	-0.67	0.68
ResidualSugar	0.00	0.00	0.23	0.00	0.23
Chlorides	-0.04	-0.04	0.42	-0.08	0.50
FreeSulfurDioxide	0.00	0.00	0.56	0.03	0.56
TotalSulfurDioxide	0.00	0.00	0.69	0.01	0.69
Density	-0.26	-0.26	-0.08	-0.05	0.21
pH	-0.01	-0.01	0.77	0.02	0.78
Sulphates	-0.01	-0.01	1.08	0.03	1.09
Alcohol	0.00	0.00	1.20	-0.03	1.24
LabelAppeal-1	0.24	0.24	1.32	0.03	1.29
LabelAppeal0	0.43	0.43	0.00	0.24	0.43
LabelAppeal1	0.56	0.56	-0.03	0.43	0.58
LabelAppeal2	0.70	0.70	0.01	0.56	0.69
AcidIndex	-0.08	-0.08	0.00	0.69	0.77
STARS1	0.77	0.77	-0.03	0.77	0.80
STARS2	1.09	1.09	0.00	1.09	1.09
STARS3	1.20	1.20	0.00	1.21	1.21
STARS4	1.33	1.33	-0.01	1.33	1.34

### 3 BUILD MODELS

#### 3.1 Poisson Regression Model

Poisson regression models were run on all 4 data preparation methods we chose using a full model approach that included all predictors in each model. The data preparation method did not seem to have any meaningful impact on the performance of the models with all four resulting in similar AIC values although the coefficients were significantly different. The comparison can be seen in tables 4 and 5.

Further models were created using the data prep strategy that involved adding the minimum value plus one to all variables that had negative values in order to remove all negative and zero values from our data.

Using the full model above as a base, a `step` function was run to reduce the number of variables in the model resulting in significantly fewer variables. The `drop1` function was then run to see if any more variables could be removed and `pH` was dropped from the model as a result.

An influence plot was also run to see if there were any influential points affecting the model. Five influential points (records 3953, 4940, 8887, 10108, and 12513) were identified and removed from the model to see if they improved the fit. The improvement was modest but noticeable so they were left out of the remaining Poisson models.

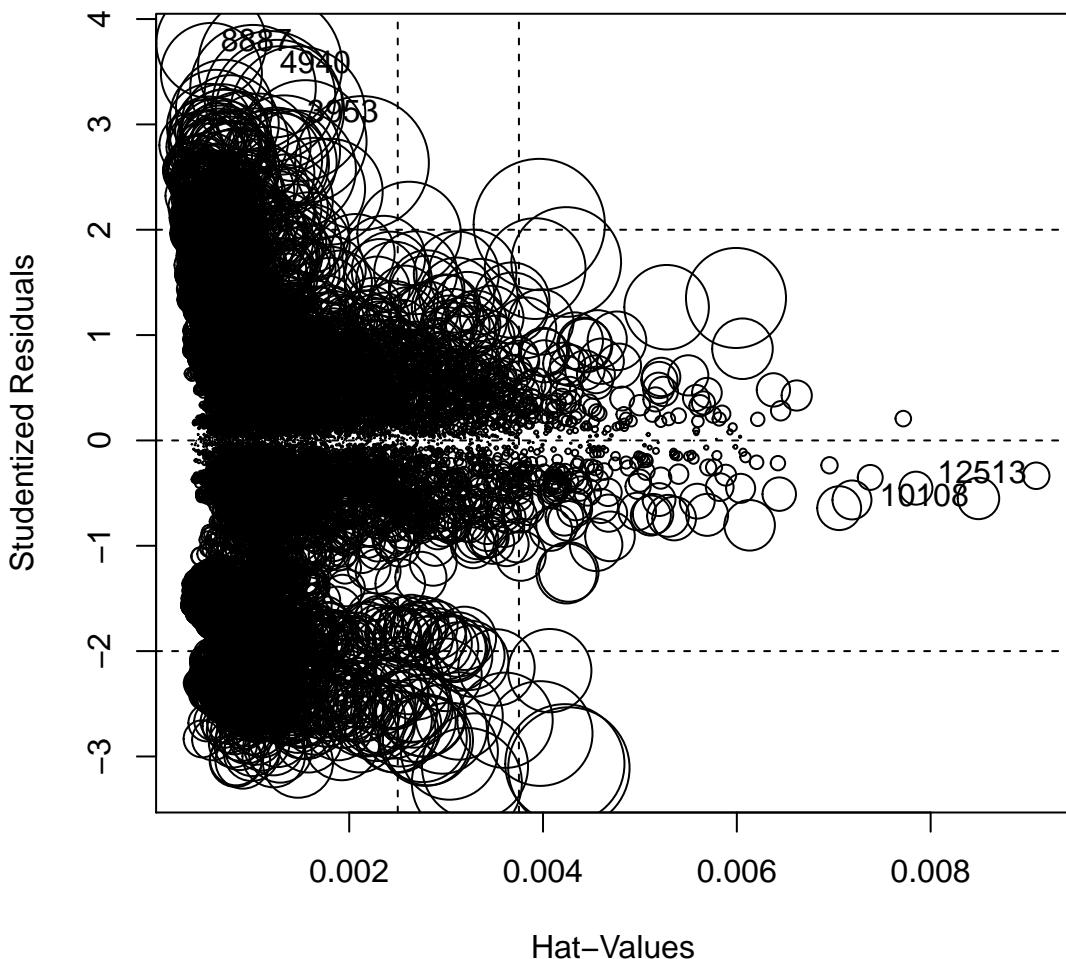


Figure 10: Poisson model influence plot

StudRes	Hat	CookD
3.1	0.00147	0.00161
3.56	0.00118	0.00183
3.77	0.000576	0.00106
-0.551	0.00849	0.000152
-0.335	0.00909	6.2e-05

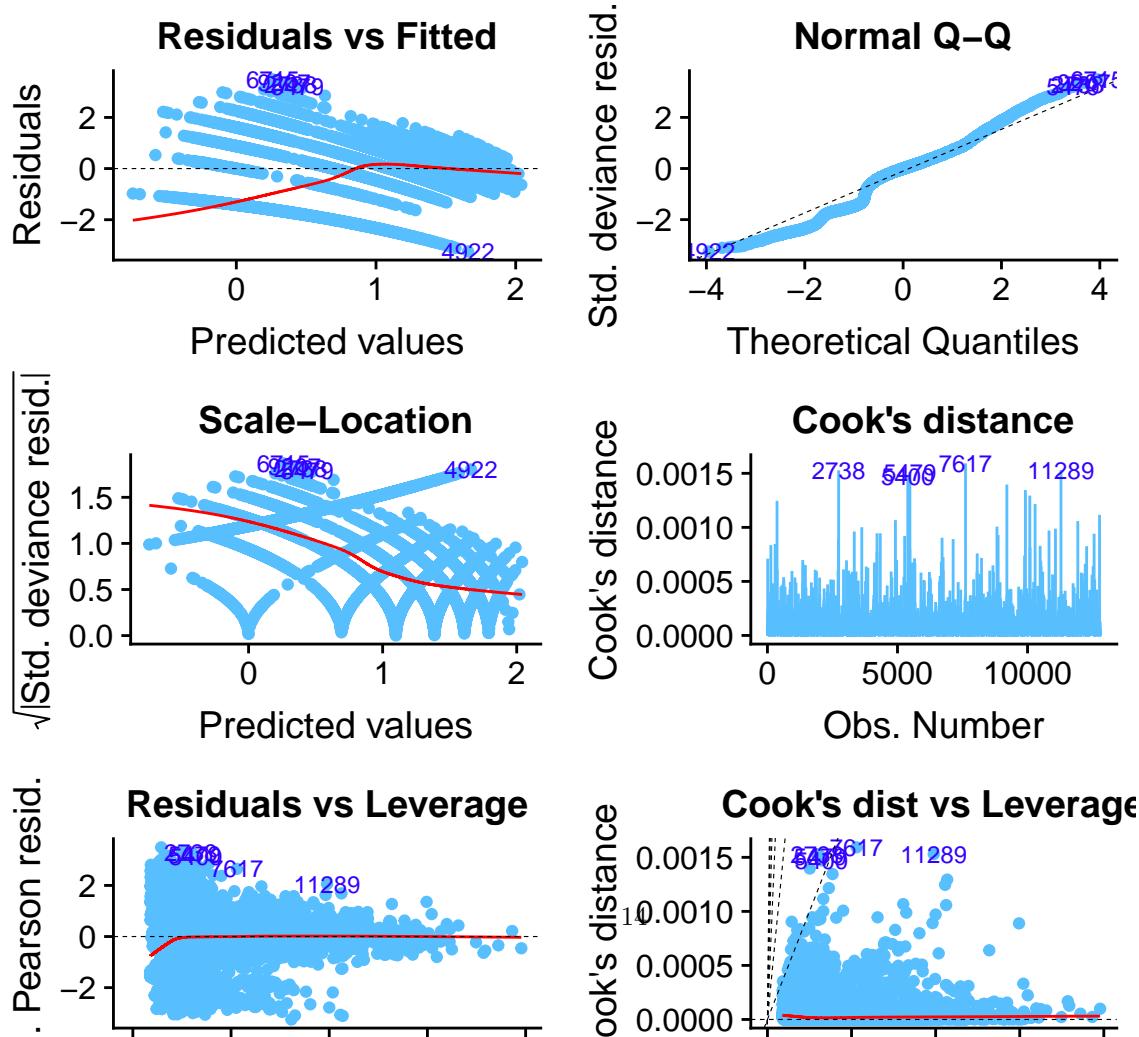
### 3.1.0.1 Base Poisson Model Statistics and Coefficients

Observations	12790
Dependent variable	TARGET
Type	Generalized linear model
Family	poisson
Link	log

$\chi^2(15)$	9235.52
Pseudo-R <sup>2</sup> (Cragg-Uhler)	0.52
Pseudo-R <sup>2</sup> (McFadden)	0.17
AIC	45559.21
BIC	45678.52

	Est.	S.E.	z val.	p	VIF
(Intercept)	0.49	0.08	5.85	0.00	NA
STARS1	0.77	0.02	39.40	0.00	1.17
STARS2	1.09	0.02	59.71	0.00	1.17
STARS3	1.21	0.02	62.93	0.00	1.17
STARS4	1.33	0.02	54.70	0.00	1.17
LabelAppeal-1	0.24	0.04	6.19	0.00	1.13
LabelAppeal0	0.42	0.04	11.47	0.00	1.13
LabelAppeal1	0.56	0.04	14.76	0.00	1.13
LabelAppeal2	0.69	0.04	16.22	0.00	1.13
AcidIndex	-0.08	0.00	-17.58	0.00	1.03
VolatileAcidity	-0.03	0.01	-4.82	0.00	1.00
TotalSulfurDioxide	0.00	0.00	3.55	0.00	1.00
Alcohol	0.00	0.00	2.83	0.00	1.01
FreeSulfurDioxide	0.00	0.00	2.67	0.01	1.00
Sulphates	-0.01	0.01	-1.89	0.06	1.00
Chlorides	-0.04	0.02	-2.24	0.03	1.00

Standard errors: MLE



### 3.1.1 Quasipoisson Model

Next a quasipoisson model was tried using all the same steps outlined for the Poisson model above that resulted in a model that had very little change and no real improvement to the model fit over the Poisson model. The standard errors of the coefficients were moderately reduced for about half the variables but significantly increased for the other half.

poisson	quasip	se.poiss	se.quasi	ratio
0.486	0.486	0.0831	0.0779	0.938
0.771	-0.0315	0.0196	0.0061	0.313
1.09	-0.0358	0.0183	0.015	0.82
1.21	0.0001	0.0192	0	0.0017
1.33	0.0001	0.0244	0	0.0009
0.235	-0.0103	0.038	0.0051	0.135
0.425	0.0039	0.0371	0.0013	0.0348
0.556	0.235	0.0377	0.0356	0.945
0.689	0.425	0.0425	0.0347	0.818
-0.0792	0.556	0.0045	0.0353	7.85
-0.0315	0.689	0.0065	0.0398	6.09
0.0001	-0.0792	0	0.0042	191
0.0039	0.771	0.0014	0.0184	13.3
0.0001	1.09	0	0.0171	500
-0.0103	1.21	0.0054	0.018	3.31
-0.0358	1.33	0.016	0.0228	1.43

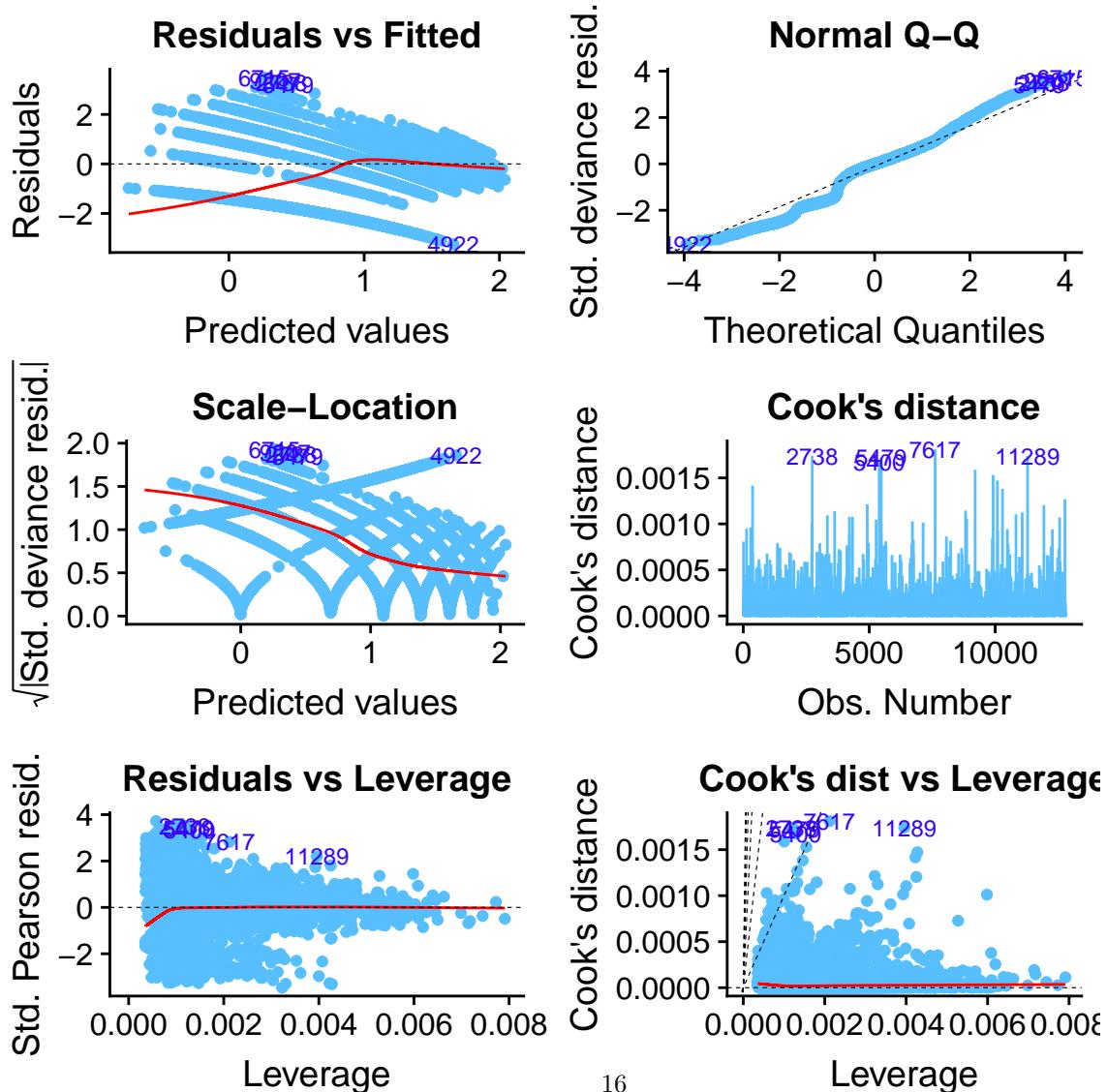
#### 3.1.1.1 Quasipoisson Model Statistics and Coefficients

Observations	12790
Dependent variable	TARGET
Type	Generalized linear model
Family	quasipoisson
Link	log

$\chi^2(15)$	9235.52
Pseudo-R <sup>2</sup> (Cragg-Uhler)	0.52
Pseudo-R <sup>2</sup> (McFadden)	0.17
AIC	NA
BIC	NA

	Est.	S.E.	t val.	p	VIF
(Intercept)	0.49	0.08	6.24	0.00	NA
VolatileAcidity	-0.03	0.01	-5.13	0.00	1.00
Chlorides	-0.04	0.01	-2.39	0.02	1.00
FreeSulfurDioxide	0.00	0.00	2.85	0.00	1.00
TotalSulfurDioxide	0.00	0.00	3.78	0.00	1.00
Sulphates	-0.01	0.01	-2.02	0.04	1.00
Alcohol	0.00	0.00	3.01	0.00	1.01
LabelAppeal-1	0.24	0.04	6.60	0.00	1.13
LabelAppeal0	0.42	0.03	12.23	0.00	1.13
LabelAppeal1	0.56	0.04	15.74	0.00	1.13
LabelAppeal2	0.69	0.04	17.29	0.00	1.13
AcidIndex	-0.08	0.00	-18.74	0.00	1.03
STARS1	0.77	0.02	42.02	0.00	1.17
STARS2	1.09	0.02	63.67	0.00	1.17
STARS3	1.21	0.02	67.11	0.00	1.17
STARS4	1.33	0.02	58.33	0.00	1.17

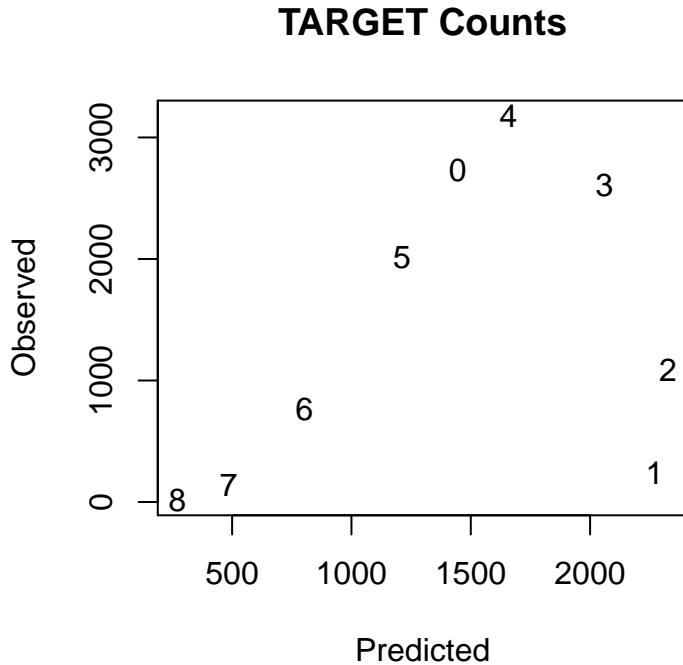
Standard errors: MLE



### 3.1.2 Hurdle and Zero-inflated Poisson Models

Hurdle and Zero-inflated models are two ways of dealing with zero-inflated data. A plot showing the expected vs. observed counts of the TARGET values was created in order to see if the data might be zero-inflated. We can clearly see in the plot that we have a much greater occurrence of zero values than we would expect while the rest of the values are much closer to our expectations. The shape of the plot does peak at 3, 4 and 5 cases and then curve downward on the right side at 1, 2, and 3 cases indicating that the observed values for the purchase of 3, 4 or 5 cases are much higher than we would expect and the observed values for the purchase of 1 or 2 cases are much lower than we would expect.

```
## integer(0)
```



Plots of the two categorical variables, STARS and LabelAppeal, when the target is zero vs. when it is not zero show a significant difference for STARS but virtually no difference for LabelAppeal. The STARS plot strongly indicates that a hurdle or zero-inflated or possibly piecewise model may be a better fit. While the LabelAppeal plot only supports the choice of hurdle and zero-inflated models.

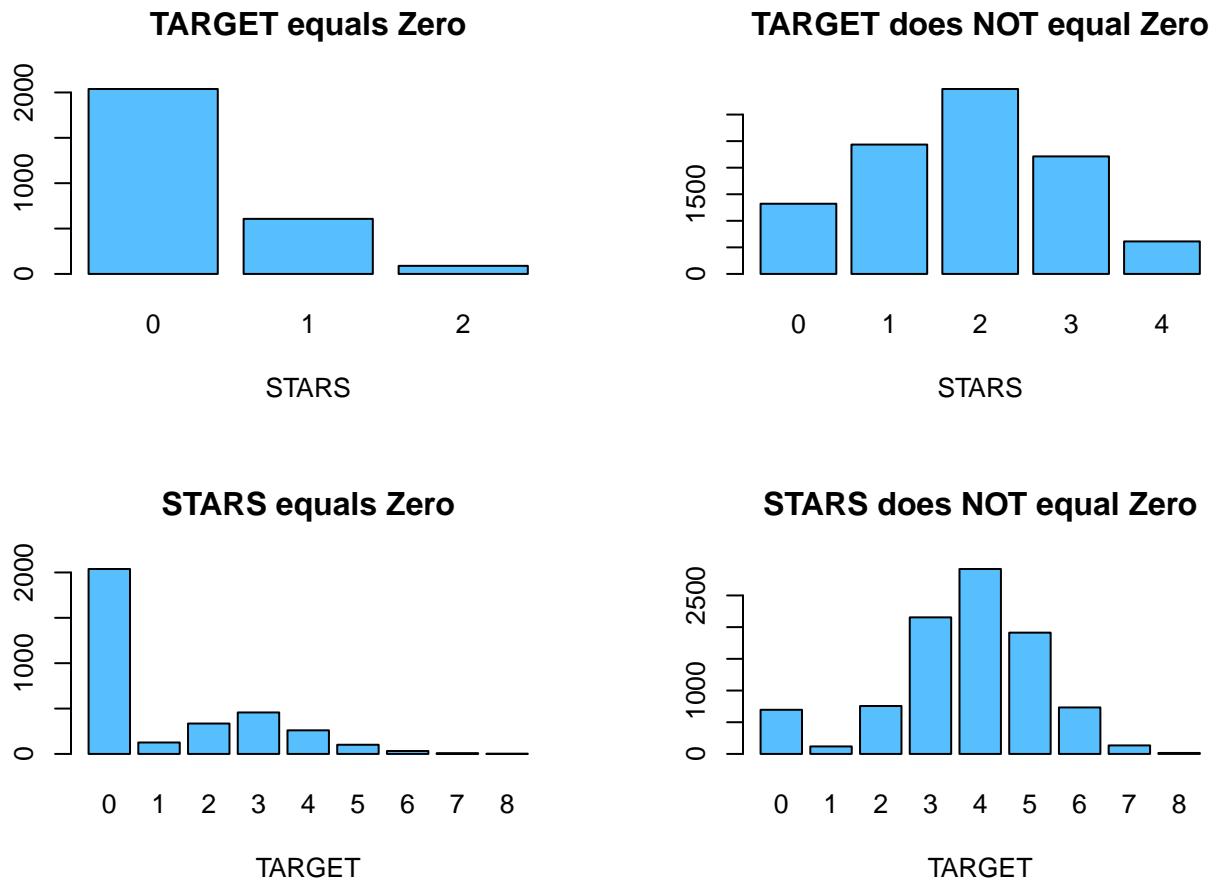


Figure 11: Difference in distributions for STARS when TARGET equals zero vs. greater than zero

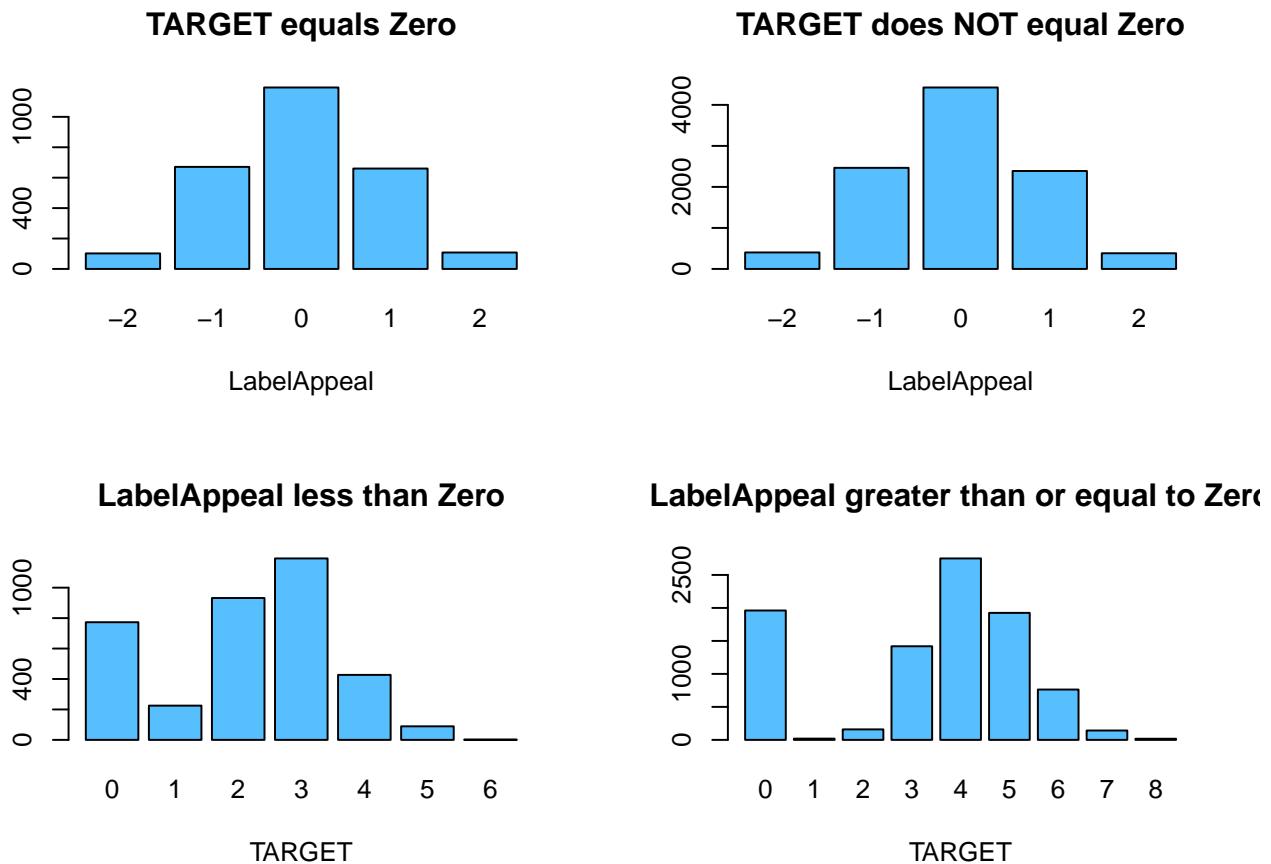
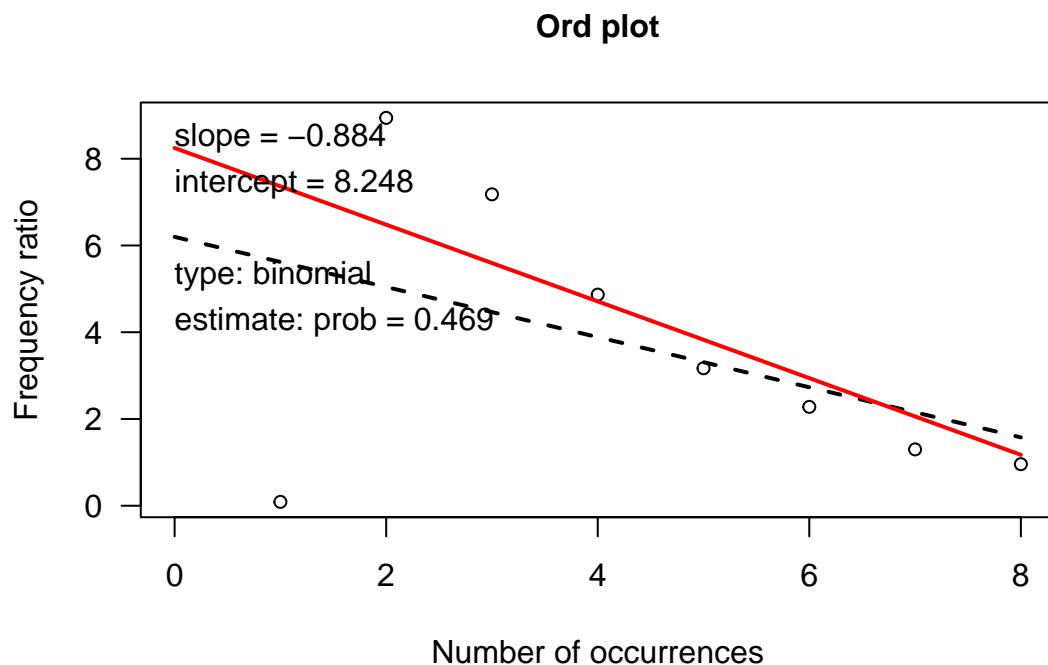
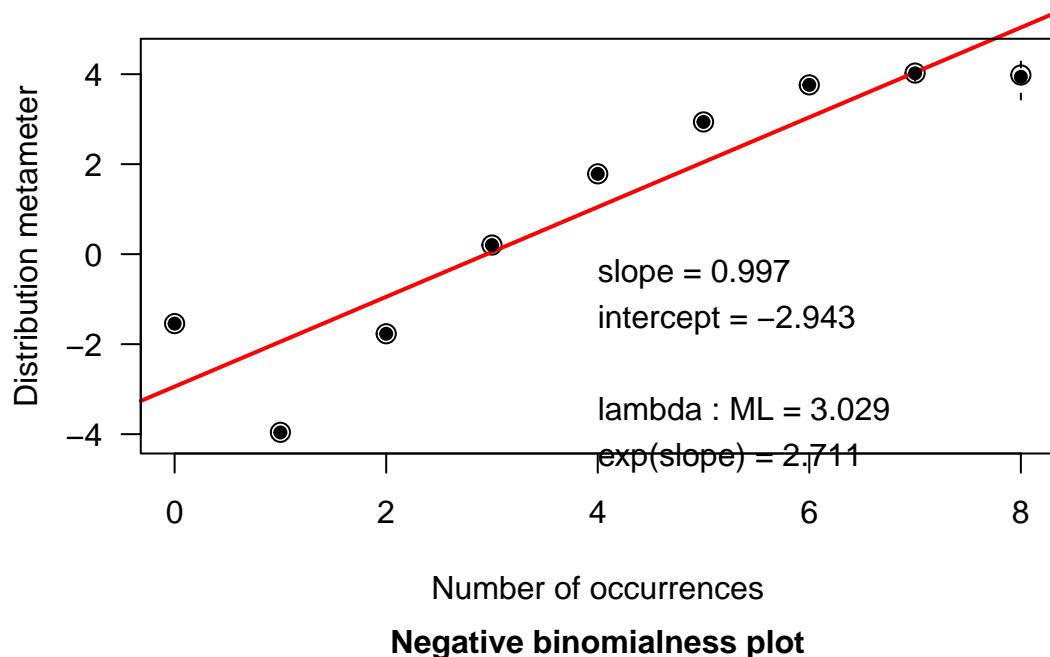


Figure 12: Difference in distributions for LabelAppeal when TARGET is less than or equal to zero vs. greater than zero

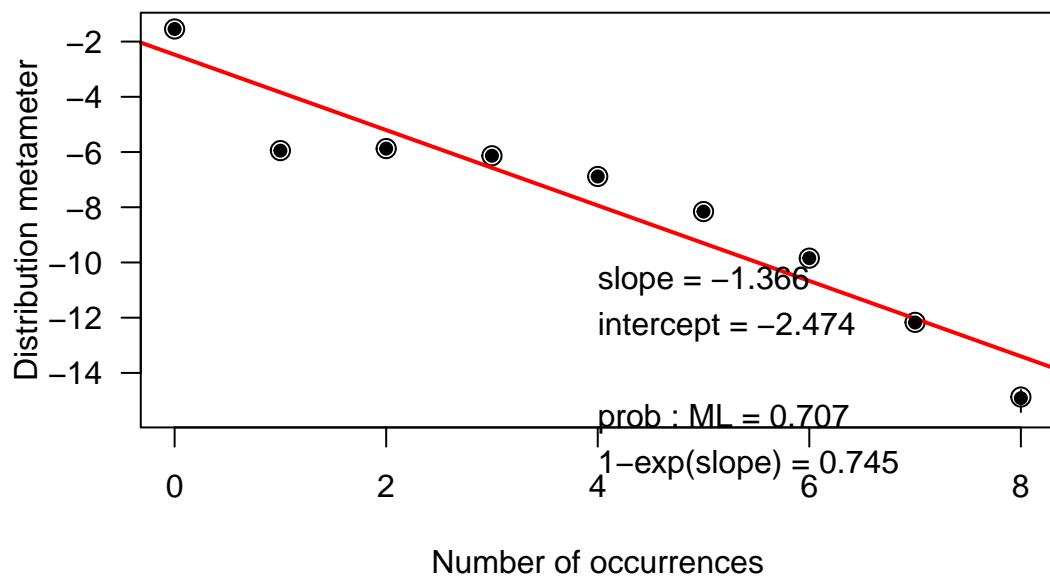
Zero-inflation is also clearly seen in the Diagnostic Distribution Plots below.

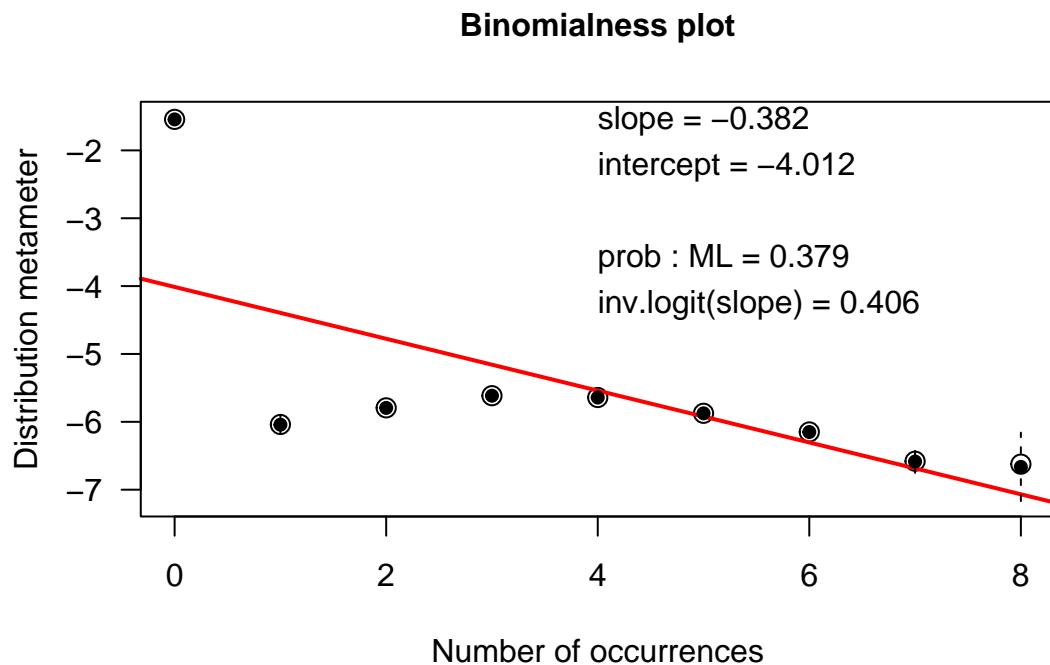


### Poissoness plot



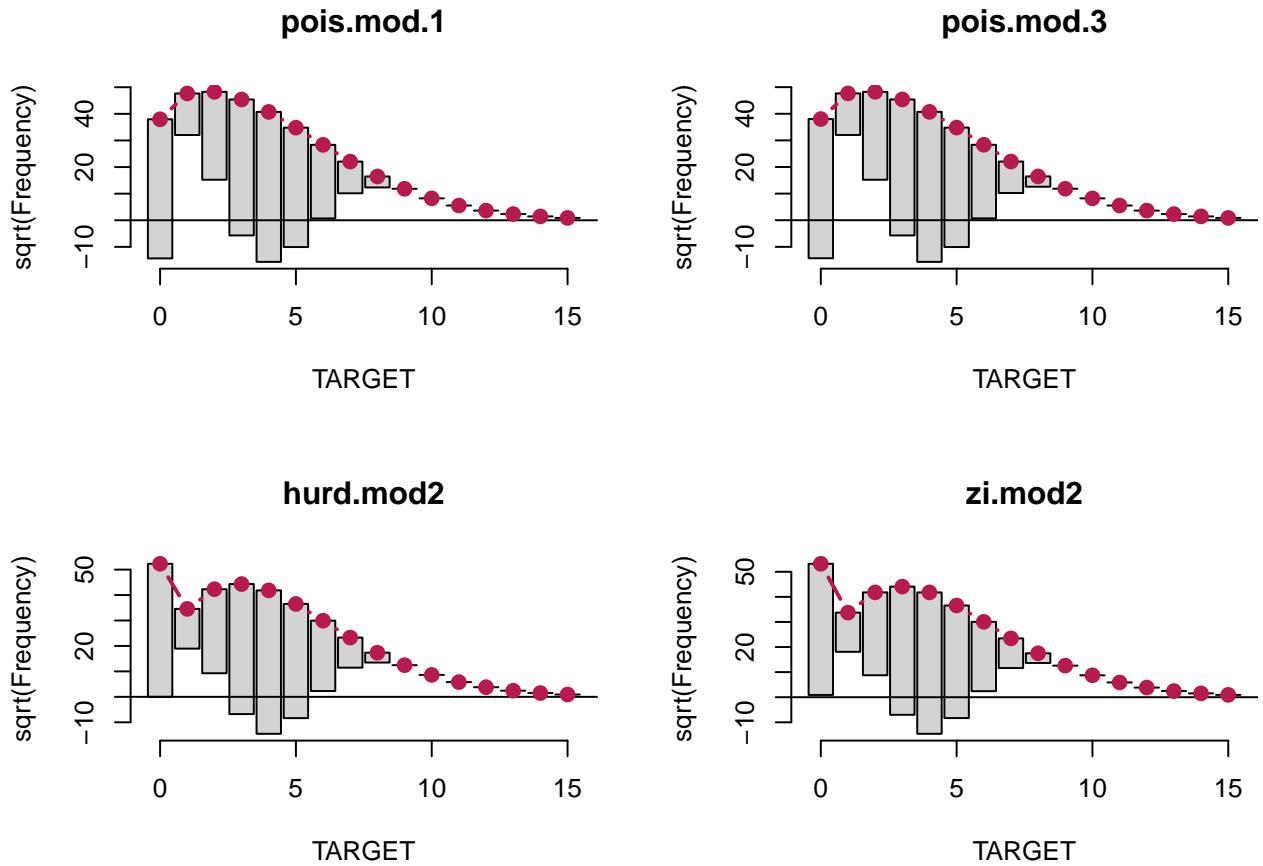
### Negative binomialness plot





Hurdle and zero-inflated models were both run and refined using backward elimination on the data with the influential points already removed. Both models showed significant improvement over the Poisson and quasipoisson models with significantly lower AIC values and rootogram plots that showed a closer fit and less dispersion, although dispersion is still an issue since there is still evidence of over and under dispersion in the hurdle and zero-inflated rootogram plots.

### 3.1.2.1 Rootogram Plots



### 3.1.2.2 Hurdle Model Statistics and Coefficients

```
##
## Call:
## hurdle(formula = TARGET ~ AcidIndex + Alcohol + LabelAppeal + STARS +
##         VolatileAcidity + FreeSulfurDioxide + TotalSulfurDioxide + pH +
##         Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS, data = minusinfluential)
##
## Pearson residuals:
##      Min     1Q   Median     3Q    Max
## -2.09275 -0.44331 -0.00245  0.39568  4.54188
##
## Count model coefficients (truncated poisson with log link):
##             Estimate Std. Error z value    Pr(>|z|)
## (Intercept) 0.33863  0.06713   5.04 0.00000046 ***
## AcidIndex   -0.01705  0.00492  -3.46 0.00053 ***
## Alcohol     0.00732  0.00145   5.07 0.00000041 ***
## LabelAppeal-1 0.53977  0.04973  10.85 < 0.0000000000000002 ***
## LabelAppeal0  0.84299  0.04880  17.27 < 0.0000000000000002 ***
## LabelAppeal1  1.03971  0.04937  21.06 < 0.0000000000000002 ***
## LabelAppeal2  1.19915  0.05324  22.53 < 0.0000000000000002 ***
## STARS1      0.05342  0.02146   2.49 0.01283 *
## STARS2      0.16856  0.02002   8.42 < 0.0000000000000002 ***
## STARS3      0.25901  0.02098  12.35 < 0.0000000000000002 ***
## STARS4      0.36350  0.02592  14.02 < 0.0000000000000002 ***
##
## Zero hurdle model coefficients (binomial with logit link):
```

```

##                                     Estimate Std. Error z value      Pr(>|z|)
## (Intercept)                 4.433024  0.399013  11.11 < 0.0000000000000002 ***
## VolatileAcidity          -0.187381  0.036488  -5.14     0.000000281567 ***
## FreeSulfurDioxide        0.000666  0.000197   3.37      0.00074 ***
## TotalSulfurDioxide       0.000819  0.000127   6.46     0.000000000103 ***
## pH                         -0.172426  0.041928  -4.11     0.000039149953 ***
## Sulphates                  -0.095360  0.030502  -3.13      0.00177 **
## Alcohol                     -0.018903  0.007666  -2.47      0.01367 *
## LabelAppeal-1              -0.483545  0.137302  -3.52      0.00043 ***
## LabelAppeal0                -0.903071  0.134048  -6.74     0.000000000016 ***
## LabelAppeal1                -1.448751  0.143581 -10.09 < 0.0000000000000002 ***
## LabelAppeal2                -1.879326  0.223186  -8.42 < 0.0000000000000002 ***
## AcidIndex                   -0.388161  0.021425 -18.12 < 0.0000000000000002 ***
## STARS1                      1.831113  0.061349  29.85 < 0.0000000000000002 ***
## STARS2                      4.268926  0.117246  36.41 < 0.0000000000000002 ***
## STARS3                      20.250040 363.224872   0.06      0.95554
## STARS4                      20.406402 695.238140   0.03      0.97658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 18
## Log-likelihood: -2.03e+04 on 27 Df

```

### 3.1.2.3 Zero-inflated Poisson Model Statistics and Coefficients

```

##
## Call:
## zeroinfl(formula = TARGET ~ AcidIndex + Alcohol + LabelAppeal +
##           STARS | VolatileAcidity + FreeSulfurDioxide + TotalSulfurDioxide +
##           pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS,
##           data = minusinfluential)
##
## Pearson residuals:
##      Min      1Q      Median      3Q      Max
## -2.24755 -0.42723  0.00257  0.38363  5.33255
##
## Count model coefficients (poisson with log link):
##                                     Estimate Std. Error z value      Pr(>|z|)
## (Intercept)                 0.47099  0.06056   7.78  0.00000000000074 ***
## AcidIndex                  -0.01960  0.00482  -4.07  0.000047666666659 ***
## Alcohol                      0.00685  0.00140   4.88  0.0000010526444344 ***
## LabelAppeal-1               0.44004  0.04128  10.66 < 0.0000000000000002 ***
## LabelAppeal0                 0.72789  0.04035  18.04 < 0.0000000000000002 ***
## LabelAppeal1                 0.91746  0.04102  22.36 < 0.0000000000000002 ***
## LabelAppeal2                 1.07369  0.04560  23.55 < 0.0000000000000002 ***
## STARS1                      0.06468  0.02118   3.05      0.0023 **
## STARS2                      0.18706  0.01981   9.44 < 0.0000000000000002 ***
## STARS3                      0.28455  0.02075  13.72 < 0.0000000000000002 ***
## STARS4                      0.38397  0.02567  14.96 < 0.0000000000000002 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                                     Estimate Std. Error z value      Pr(>|z|)
## (Intercept)                 -6.294463  0.569851 -11.05 < 0.0000000000000002 ***
## VolatileAcidity            0.198885  0.043469   4.58     0.0000047553093 ***

```

```

## FreeSulfurDioxide -0.000818 0.000238 -3.44          0.00058 ***
## TotalSulfurDioxide -0.000912 0.000152 -5.99          0.0000000021358 ***
## pH                  0.198507 0.049925 3.98          0.0000700470654 ***
## Sulphates           0.116125 0.036433 3.19          0.00144 **
## Alcohol              0.025501 0.009232 2.76          0.00574 **
## LabelAppeal-1       1.497177 0.329672 4.54          0.0000055878285 ***
## LabelAppeal0         2.254987 0.326985 6.90          0.0000000000053 ***
## LabelAppeal1         2.968020 0.332370 8.93 < 0.000000000000002 ***
## LabelAppeal2         3.492351 0.385069 9.07 < 0.000000000000002 ***
## AcidIndex            0.430443 0.025686 16.76 < 0.000000000000002 ***
## STARS1               -2.091442 0.076112 -27.48 < 0.000000000000002 ***
## STARS2               -5.731503 0.322363 -17.78 < 0.000000000000002 ***
## STARS3               -20.250172 339.002169 -0.06          0.95237
## STARS4               -20.406439 639.781273 -0.03          0.97456
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 36
## Log-likelihood: -2.03e+04 on 27 Df

```

## 3.2 Negative binomial regression models

Negative binomial regression is an alternative when there is over dispersion ( $\text{var}(Y_i) > E(Y_i)$ ).

“A Poisson distribution is parameterized by  $\lambda$ , which happens to be both its mean and variance. While convenient, it’s not often realistic. A distribution of counts will usually have a variance that’s not equal to its mean. When we see this happen with data that we assume is Poisson distributed, we say we have under- or over dispersion, depending on if the variance is smaller or larger than the mean. Performing Poisson regression on count data that exhibits this behavior results in a model that doesn’t fit well.”

“One approach that addresses this issue is the Negative Binomial Regression. The negative binomial distribution describes the probabilities of the occurrence of whole numbers greater than or equal to 0. Unlike the Poisson distribution, the variance and the mean are not equivalent. This suggests it might serve as a useful approximation for modeling counts with variability different from its mean. The variance of a negative binomial distribution is a function of its mean and has an additional parameter  $k$  called the dispersion parameter. Say our count is a random variable  $Y$  from a negative binomial distribution, when the variance of  $Y$  is:”

$$\text{var}(Y) = \mu + \mu^2 lk$$

“As the dispersion parameter gets larger and larger, the variance converges to the same value as the mean, and the negative binomial turns into a Poisson distribution.”

```

## [1] "Mean of Target: 3"
## [1] "Variance of Target: 3.711"

```

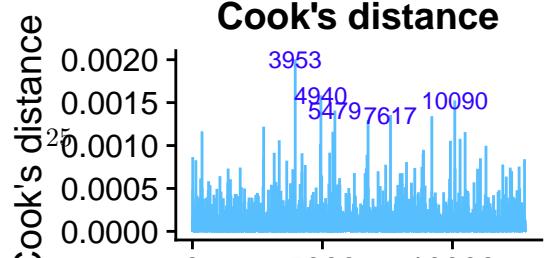
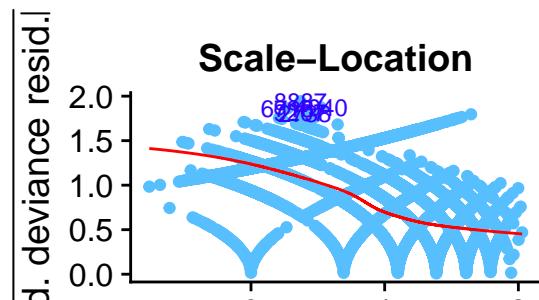
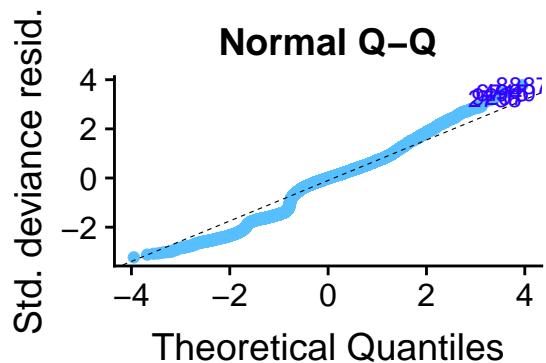
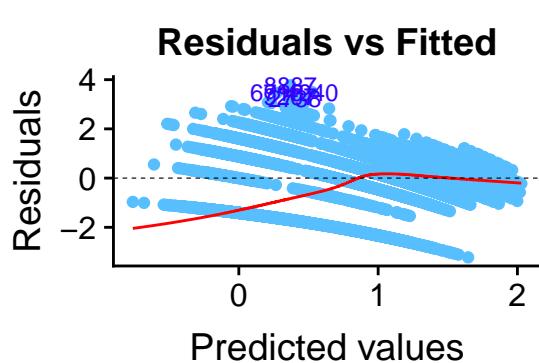
It appears that there is a slight over dispersion with the variance greater than the mean as shown above.

### 3.2.1 Negative binomial regression model 1

First negative binomial model is using all variables and selecting predictors using stepAIC.

Observations	12795			
Dependent variable	TARGET			
Type	Generalized linear model			
Family	Negative Binomial(40840)			
Link	log			
$\chi^2()$	0.50	0.16	45611.20	45775.25
Pseudo-R <sup>2</sup> (Cragg-Uhler)	0.50	0.16	45611.20	45775.25
Pseudo-R <sup>2</sup> (McFadden)	0.50	0.16	45611.20	45775.25
AIC	0.50	0.16	45611.20	45775.25
BIC	0.50	0.16	45611.20	45775.25
	Est.	S.E.	z val.	p
(Intercept)	0.70	0.20	3.53	0.00
Density	-0.27	0.19	-1.39	0.17
pH	-0.01	0.01	-1.59	0.11
Alcohol	0.00	0.00	2.83	0.00
LabelAppeal1	0.23	0.04	6.18	0.00
LabelAppeal2	0.42	0.04	11.47	0.00
LabelAppeal3	0.56	0.04	14.76	0.00
LabelAppeal4	0.69	0.04	16.36	0.00
AcidIndex	-0.08	0.00	-17.37	0.00
STARS1	0.77	0.02	39.18	0.00
STARS2	1.08	0.02	59.38	0.00
STARS3	1.20	0.02	62.65	0.00
STARS4	1.32	0.02	54.47	0.00
FixedAcidity	-0.00	0.00	-0.05	0.96
VolatileAcidity	-0.03	0.00	-5.28	0.00
CitricAcid	0.01	0.00	1.23	0.22
ResidualSugar	0.00	0.00	0.30	0.76
Chlorides	-0.03	0.01	-2.62	0.01
FreeSulfurDioxide	0.00	0.00	2.58	0.01
TotalSulfurDioxide	0.00	0.00	3.68	0.00
Sulphates	-0.01	0.00	-2.15	0.03

Standard errors: MLE



It is shown that there is an issue with non-constant variance and long tails in the qqplot. The best performing dataset was the one with negative values arithmetically scaled from lower bound of  $IQR^*1.5$  to 0, and lesser values dropped. The negative binomial model appears to have many multiple statistically significant values.

### 3.2.2 Negative binomial regression model 2

Observations	12795			
Dependent variable	TARGET			
Type	Generalized linear model			
Family	Negative Binomial(40838)			
Link	log			
$\chi^2()$	0.50	0.16	45606.79	45748.46
Pseudo-R <sup>2</sup> (Cragg-Uhler)	0.50	0.16	45606.79	45748.46
Pseudo-R <sup>2</sup> (McFadden)	0.50	0.16	45606.79	45748.46
AIC	0.50	0.16	45606.79	45748.46
BIC	0.50	0.16	45606.79	45748.46

	Est.	S.E.	z val.	p
(Intercept)	0.71	0.20	3.56	0.00
Density	-0.27	0.19	-1.40	0.16
pH	-0.01	0.01	-1.58	0.11
Alcohol	0.00	0.00	2.84	0.00
LabelAppeal1	0.23	0.04	6.18	0.00
LabelAppeal2	0.43	0.04	11.47	0.00
LabelAppeal3	0.56	0.04	14.77	0.00
LabelAppeal4	0.69	0.04	16.36	0.00
AcidIndex	-0.08	0.00	-17.53	0.00
STARS1	0.77	0.02	39.20	0.00
STARS2	1.08	0.02	59.42	0.00
STARS3	1.20	0.02	62.68	0.00
STARS4	1.32	0.02	54.50	0.00
VolatileAcidity	-0.03	0.00	-5.29	0.00
Chlorides	-0.03	0.01	-2.63	0.01
FreeSulfurDioxide	0.00	0.00	2.58	0.01
TotalSulfurDioxide	0.00	0.00	3.70	0.00
Sulphates	-0.01	0.00	-2.15	0.03

Standard errors: MLE

```
## $residual.deviance
## [1] 13624
##
## $residual.degrees.of.freedom
## [1] 12774
##
## $chisq.p.value
## [1] 0.000000094
```

```

## $residual.deviance
## [1] 13626
##
## $residual.degrees.of.freedom
## [1] 12777
##
## $chisq.p.value
## [1] 0.000000099

```

There is no significant improvement after using `stepAIC` to select the needed predictors. The reason to use Zero Dispersion Counts model is due to an inflated number of zeros in our counts target.

### 3.3 Linear regression models

Finally, there was the need to explore the efficiency of Multiple Linear Regression in determining the `TARGET` variable for the dataset. Both models created utilized the same factors for determining the `TARGET` in slightly different ways.

#### 3.3.1 Linear regression model 1

For the first model, it was previously discovered there was a definite interaction between the `TARGET` and the variables `AcidIndex`, `LabelAppeal`, and `STARS`. The latter of these two variables are factors and as a result have a lot of potential interactions. To start with, just the variables themselves were examined as potential influencers for determining the `TARGET`.

Observations	12795
Dependent variable	TARGET
Type	OLS linear regression

F(9,12785)	1648.19
R <sup>2</sup>	0.54
Adj. R <sup>2</sup>	0.54

	Est.	S.E.	t val.	p
(Intercept)	2.10	0.09	22.48	0.00
AcidIndex	-0.21	0.01	-23.01	0.00
LabelAppeal-1	0.36	0.06	5.68	0.00
LabelAppeal0	0.82	0.06	13.41	0.00
LabelAppeal1	1.29	0.06	20.04	0.00
LabelAppeal2	1.87	0.08	22.14	0.00
STARS1	1.38	0.03	41.72	0.00
STARS2	2.42	0.03	75.50	0.00
STARS3	2.99	0.04	80.75	0.00
STARS4	3.68	0.06	62.09	0.00

Standard errors: OLS

This model clearly shows the influence each variable has on the `TARGET`, although the efficiency comes into question. The  $R^2$  is less-than-stellar and the same can be said of the Adjusted  $R^2$ . Additionally, this model has an AIC of 43254.42 and a BIC of 43336.44. It is not the best model that has been seen so far, but it is the first multiple linear regression model to work with.

Outside of its relatively atrocious performance in general, as a model, it is actually decent.

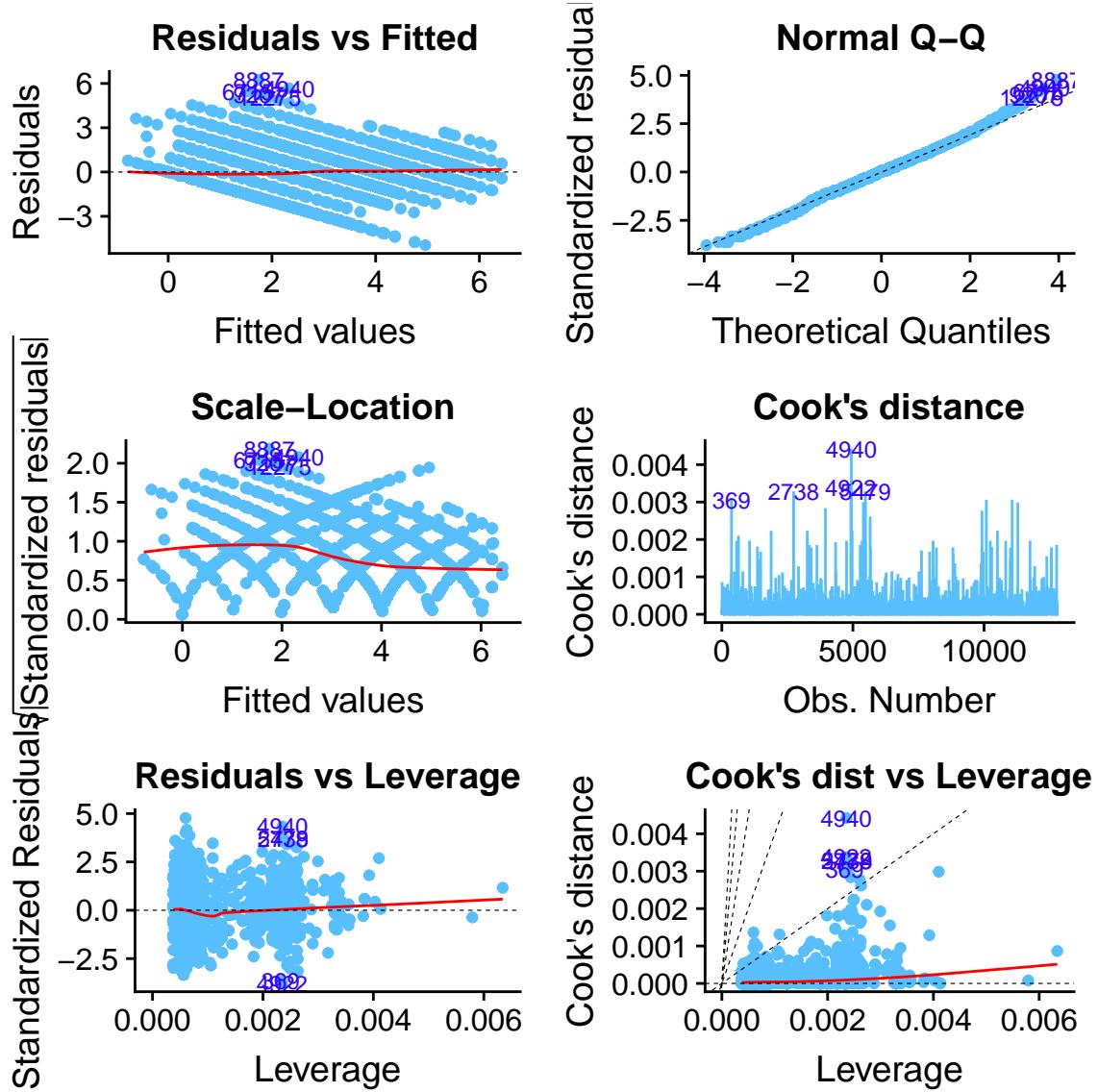


Figure 13: First Multiple Linear Regression Model's Plots

The residuals are nearly even across the line for the Residuals vs. Fitted plot, and the Normal Q-Q plot shows the residuals are following neatly across the line, indicating they are normally distributed. The Scale-Location plot shows an interesting distribution, but on the whole, the line is nearly horizontal but the points are not necessarily random. The Cook's Distance plot shows there are a number of observed outliers that become more difficult to observe in the Residuals vs Leverage plot, and really stand out in the Cook's Distance vs Leverage plot. This, together, indicates a good model, but the numbers and Scale Location plot show this good model is just not good enough.

### 3.3.1.1 Confusion Matrix

<b>-2</b>	<b>-1</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
-2	0	0	5	0	0	1	0	0	0	0
-1	0	0	195	44	42	18	3	1	0	0
0	0	0	1352	84	290	411	176	43	3	0
1	0	0	742	92	400	494	225	83	27	6
2	0	0	367	23	281	926	829	236	22	2
3	0	0	69	1	77	735	1591	888	200	8
4	0	0	4	0	1	26	339	661	361	69
5	0	0	0	0	0	0	14	102	152	11
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0

Figure 14: First Multiple Linear Regression Model's Confusion Matrix

Another efficient way to consider the accuracy of a model is to look at its confusion matrix. Above are a variety of possible values for TARGET based off of the predictions and the actual TARGET values itself. This first model overall does not excel at properly predicting the values, as it did not guess a single value to be 6, 7, or 8, although there were observations of the TARGET variable with those values. Overall, the accuracy is 0.23, which shows just how little this model should be trusted.

### 3.3.2 Linear regression model 2

The second model sought to expand the reach of the first; that is, in addition to looking at each individual variable combined as a group to see how it implies the TARGET, looking also at how each of those variables interact with each other.

Observations	12795
Dependent variable	TARGET
Type	OLS linear regression
<hr/>	
F(47,12747)	355.38
R <sup>2</sup>	0.57
Adj. R <sup>2</sup>	0.57

Once again, this is another model lacking in efficiency. The R<sup>2</sup> is on par with the Adjusted R<sup>2</sup>, both which are only nominally better than the first model's. This model's AIC is 42470.87 and its BIC is 42836.26. It is better than the first model, but it still is nowhere near the best model of those investigated.

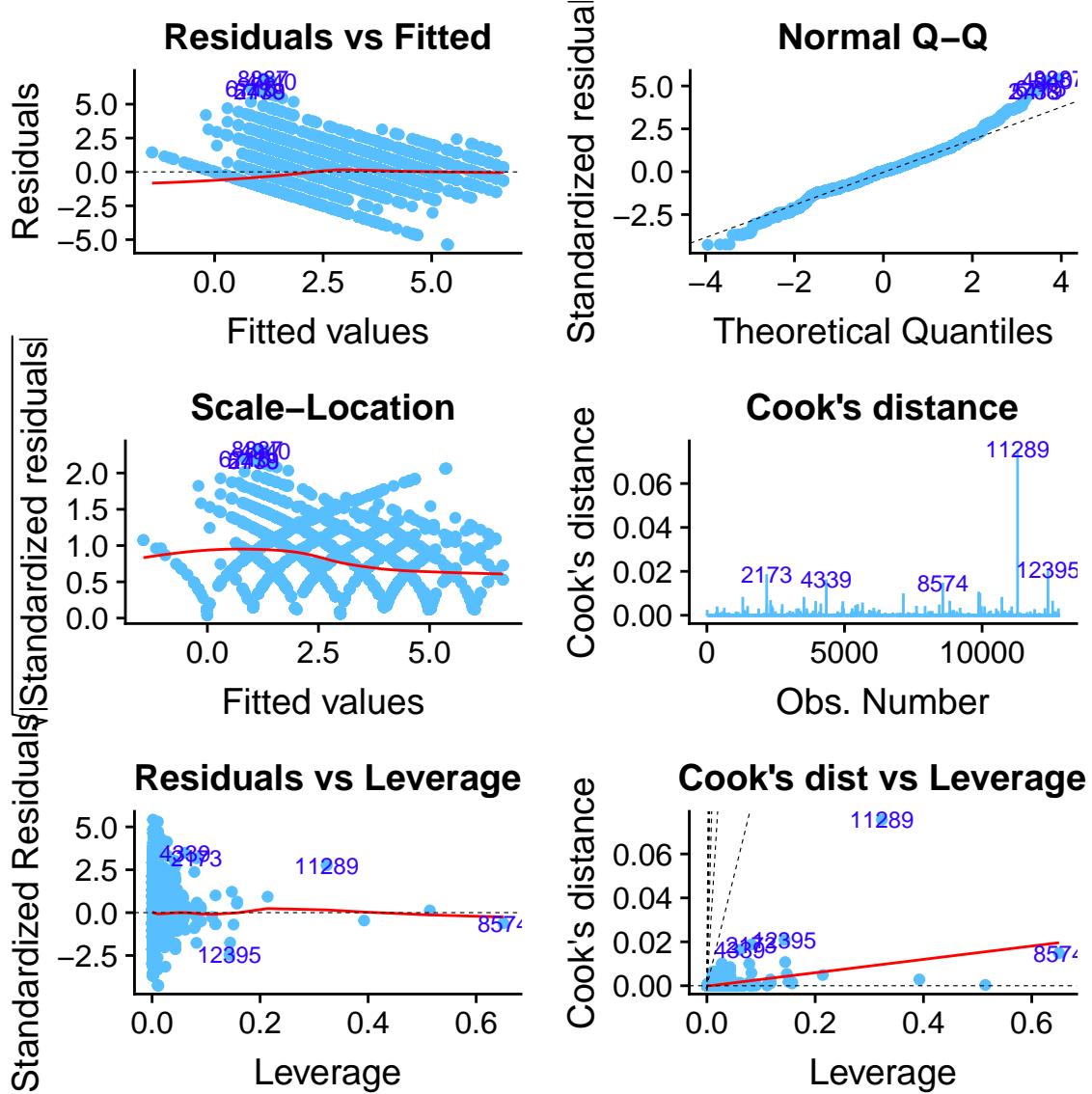


Figure 15: Second Multiple Linear Regression Model's Plots

Residuals are nearly even across the line for the Residuals vs. Fitted plot, and the Normal Q-Q plot shows the residuals are following the line less precisely as in the first model. The line for the Scale-Location plot shows less consistency while maintaining its horizontal pattern, and the points are not scattered randomly across the line. The Cook's Distance plot shows there are fewer observed outliers that are still obvious in the Residuals vs Leverage plot, and really stand out in the Cook's Distance vs Leverage plot. Combined, this further drives home the point that this model is decent, but still not a good choice.

### 3.3.2.1 Confusion Matrix

<b>-2</b>	<b>-1</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
-2	0	0	13	0	0	0	0	0	0	0
-1	0	0	153	1	1	7	2	1	0	0
0	0	0	1548	134	312	330	165	67	22	6
1	0	0	635	92	453	608	249	68	15	2
2	0	0	325	15	261	889	737	203	17	0
3	0	0	42	2	62	748	1490	670	113	3
4	0	0	18	0	2	29	508	827	365	41
5	0	0	0	0	0	0	26	178	230	87
6	0	0	0	0	0	0	0	0	3	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0

Figure 16: Second Multiple Linear Regression Model's Confusion Matrix

This model is nearly as abysmal at its assigning the TARGET variable as the first, with an accuracy of 0.23. This model is definitely better than the first multiple linear regression model, but it might be less trustworthy than rolling a multi-sided dice with each face having one of the possible TARGET values.

## 4 SELECT MODELS

### 4.1 Comparison of models

Assessing the fit of a count regression model is not necessarily straightforward; often we just look at residuals, which invariably contain patterns of some form due to the discrete nature of the observations, or we plot observed versus fitted values as a scatter plot.

Kleiber and Zeileis (2016) <https://arxiv.org/abs/1605.01311> proposes `rootogram` as an improved approach to the assessment of fit of a count regression model. The paper is illustrated using R and the authors' `countreg` package.

Rootograms are calculated using the `rootogram()` function. You can provide the observed and expected (given the model) counts as arguments to `rootogram()` or, most usefully for our purposes, a fitted count model object from which the relevant values will be extracted. `rootogram()` knows about `glm`, `gam`, `gamlss`, `hurdle`, and `zeroinfl` objects at the time of writing.

Three different kinds of rootograms are discussed in the paper

- Standing,
- Hanging, and
- Suspended.

Kleiber and Zeileis (2016) recommend hanging or suspended rootograms. Which type of rootogram is produced is controlled via argument `style`.

We will look at six different models, two Poisson models, two negative-binomial models and an ols regression thrown in for good measure.

Both the Poisson-Logit Hurdle Regression and the zero-inflated Poisson are very close in log likelihoods and BIC's.

The Poisson-Logit Hurdle Regression provides a closer fit to the observed than does the other models. The hurdle model is a modified count model in which there are two processes, one generating the zeros and one generating the positive values.

The Poisson-logit hurdle model is clearly the best choice here. The results for this model are given below.

```
##  
## Call:  
## hurdle(formula = TARGET ~ AcidIndex + Alcohol + LabelAppeal + STARS |  
##     VolatileAcidity + FreeSulfurDioxide + TotalSulfurDioxide + pH +  
##     Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS, data = minusinfluential)  
##  
## Pearson residuals:  
##      Min      1Q   Median      3Q      Max  
## -2.09275 -0.44331 -0.00245  0.39568  4.54188  
##  
## Count model coefficients (truncated poisson with log link):  
##              Estimate Std. Error z value    Pr(>|z|)  
## (Intercept)  0.33863   0.06713   5.04    0.00000046 ***  
## AcidIndex   -0.01705   0.00492  -3.46    0.00053 ***  
## Alcohol     0.00732   0.00145   5.07    0.00000041 ***  
## LabelAppeal-1 0.53977   0.04973  10.85 < 0.0000000000000002 ***  
## LabelAppeal0  0.84299   0.04880  17.27 < 0.0000000000000002 ***  
## LabelAppeal1  1.03971   0.04937  21.06 < 0.0000000000000002 ***  
## LabelAppeal2  1.19915   0.05324  22.53 < 0.0000000000000002 ***  
## STARS1      0.05342   0.02146   2.49    0.01283 *
```

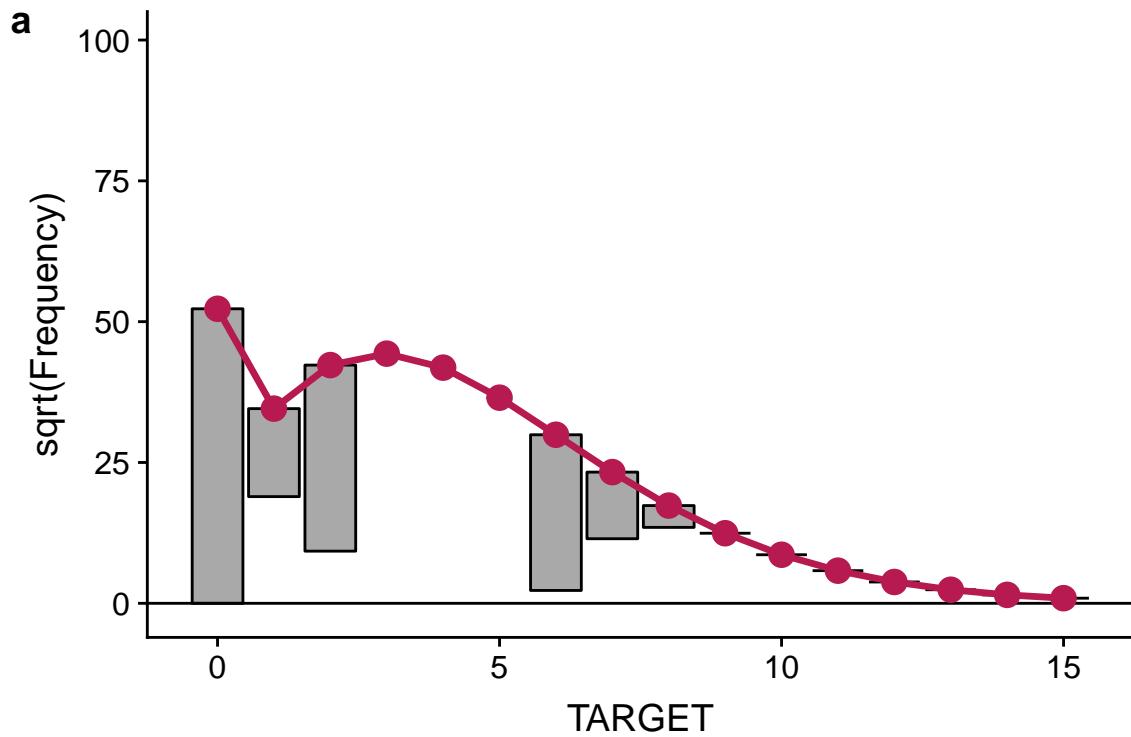
```

## STARS2      0.16856   0.02002    8.42 < 0.0000000000000002 ***
## STARS3      0.25901   0.02098   12.35 < 0.0000000000000002 ***
## STARS4      0.36350   0.02592   14.02 < 0.0000000000000002 ***
## Zero hurdle model coefficients (binomial with logit link):
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) 4.433024  0.399013 11.11 < 0.0000000000000002 ***
## VolatileAcidity -0.187381  0.036488 -5.14    0.000000281567 ***
## FreeSulfurDioxide 0.000666  0.000197  3.37    0.00074 ***
## TotalSulfurDioxide 0.000819  0.000127  6.46    0.000000000103 ***
## pH          -0.172426  0.041928 -4.11    0.000039149953 ***
## Sulphates   -0.095360  0.030502 -3.13    0.00177 **
## Alcohol     -0.018903  0.007666 -2.47    0.01367 *
## LabelAppeal-1 -0.483545  0.137302 -3.52    0.00043 ***
## LabelAppeal0  -0.903071  0.134048 -6.74    0.000000000016 ***
## LabelAppeal1  -1.448751  0.143581 -10.09 < 0.0000000000000002 ***
## LabelAppeal2  -1.879326  0.223186 -8.42 < 0.0000000000000002 ***
## AcidIndex    -0.388161  0.021425 -18.12 < 0.0000000000000002 ***
## STARS1       1.831113  0.061349 29.85 < 0.0000000000000002 ***
## STARS2       4.268926  0.117246 36.41 < 0.0000000000000002 ***
## STARS3      20.250040 363.224872  0.06    0.95554
## STARS4      20.406402 695.238140  0.03    0.97658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 18
## Log-likelihood: -2.03e+04 on 27 Df

```

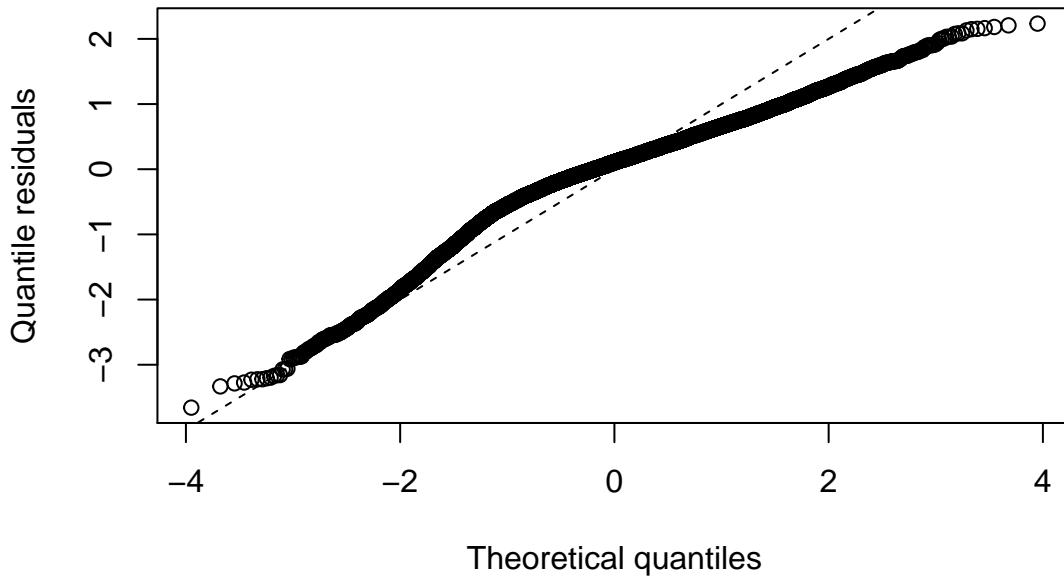
## 4.2 Diagnostic plots

Rootogram as an improved approach to the assessment of fit of a count regression model. Expected counts, given the model, are shown by the thick red line, and observed counts are shown as bars, which in a hanging rootogram are show hanging from the red line of expected count.



As a final check we can also look at the Q-Q plot of the quantile residuals in the hurdle model. These look fairly normal and show no suspicious departures from the model.

### Q-Q plot of the Poisson–Logit Hurdle Regression



### 4.3 Prediction

We ran predictions on our final model and plotted the distribution next to the distribution from our target in the training data set to compare.

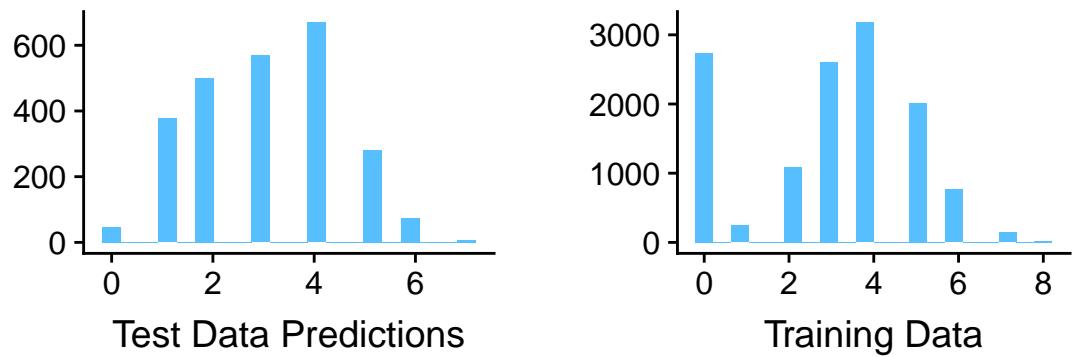


Figure 17: Predictions vs. training data

## 5 Appendix

The appendix is available as script.R file in project5\_wine folder.

[https://github.com/betsyrosalen/DATA\\_621\\_Business\\_Analyt\\_and\\_Data\\_Mining](https://github.com/betsyrosalen/DATA_621_Business_Analyt_and_Data_Mining)

```
if (!require('AICcmodavg')) (install.packages('AICcmodavg'))
if (!require('AER')) (install.packages('AER'))
if (!require('AICcmodavg')) (install.packages('AICcmodavg'))
if (!require('Amelia')) (install.packages('Amelia'))
if (!require('bookdown')) (install.packages('bookdown'))
if (!require('captioner')) (install.packages('captioner'))
if (!require('car')) (install.packages('car'))
if (!require('caret')) (install.packages('caret'))
if (!require('corrplot')) (install.packages('corrplot'))
if (!require('countreg')) (install.packages('countreg'))
if (!require('cowplot')) (install.packages('cowplot'))
if (!require('data.table')) (install.packages('data.table'))
if (!require('DataExplorer')) (install.packages('DataExplorer'))
if (!require('faraway')) (install.packages('faraway'))
if (!require('forcats')) (install.packages('forcats'))
if (!require('forecast')) (install.packages('forecast'))
if (!require('ggfortify')) (install.packages('ggfortify'))
if (!require('ggplot2')) (install.packages('ggplot2'))
if (!require('gridExtra')) (install.packages('gridExtra'))
if (!require('huxtable')) (install.packages('huxtable'))
if (!require('jtools')) (install.packages('jtools'))
if (!require('kableExtra')) (install.packages('kableExtra'))
if (!require('MASS')) (install.packages('MASS'))
if (!require('matrixStats')) (install.packages('matrixStats'))
if (!require('mice')) (install.packages('mice'))
if (!require('psych')) (install.packages('psych'))
if (!require('pROC')) (install.packages('pROC'))
if (!require('pscl')) (install.packages('pscl'))
if (!require('stats')) (install.packages('stats'))
if (!require('tidyverse')) (install.packages('tidyverse'))
if (!require('vcd')) (install.packages('vcd'))
if (!require('VIM')) (install.packages('VIM'))

# load data
train <- read.csv ('https://raw.githubusercontent.com/betsyrosalen/DATA_621_Business_Analyt_and_Data_Mining/main/winequality-red.csv',
                   stringsAsFactors = F, header = T)
test <- read.csv('https://raw.githubusercontent.com/betsyrosalen/DATA_621_Business_Analyt_and_Data_Mining/main/winequality-white.csv',
                 stringsAsFactors = F, header = T)

# remove index
train$INDEX <- NULL
test$INDEX <- NULL

vars <- rbind(c('TARGET','Number of Cases Purchased','count response'),
              c('AcidIndex','Method of testing total acidity by using a weighted avg','continuous numerical predictor'),
              c('Alcohol','Alcohol Content','continuous numerical predictor'),
              c('Chlorides','Chloride content of wine','continuous numerical predictor'),
              c('CitricAcid','Citric Acid Content','continuous numerical predictor'),
              c('Density','Density of Wine','continuous numerical predictor'),
```

```

c('FixedAcidity','Fixed Acidity of Wine','continuous numerical predictor'),
c('FreeSulfurDioxide','Sulfur Dioxide content of wine','continuous numerical predictor'),
c('LabelAppeal','Marketing Score indicating the appeal of label design','categorical predictor'),
c('ResidualSugar','Residual Sugar of wine','continuous numerical predictor'),
c('STARS','Wine rating by a team of experts. 4 = Excellent, 1 = Poor','categorical predictor'),
c('Sulphates','Sulfate content of wine','continuous numerical predictor'),
c('TotalSulfurDioxide','Total Sulfur Dioxide of Wine','continuous numerical predictor'),
c('VolatileAcidity','Volatile Acid content of wine','continuous numerical predictor'),
c('pH','pH of wine','continuous numerical predictor') )

colnames(vars) <- c('VARIABLE', 'DEFINITION', 'TYPE')

# Summary Statistics

train_num <- train[, c( 'AcidIndex', 'Alcohol', 'Density',
                      'Sulphates', 'pH', 'TotalSulfurDioxide', 'FreeSulfurDioxide', 'Chlorides',
                      'ResidualSugar', 'CitricAcid', 'VolatileAcidity', 'FixedAcidity', 'TARGET')]

train_cat <- train[, c('STARS', 'LabelAppeal')]
train_cat$STARS <- as.factor(train_cat$STARS )
train_cat$LabelAppeal <- as.factor(train_cat$LabelAppeal )

train_num_stats <- describe(train_num) [,c(2,8,3,5,9,4)]
train_cat_stats <- summary(train_cat[, c('STARS', 'LabelAppeal')])

# Data Distribution
hist <- train %>%
  gather() %>%
  ggplot(aes(value)) +
  facet_wrap(~ key, scales = "free") +
  geom_histogram(fill = "#58BFFF") +
  xlab("") +
  ylab("") +
  theme(panel.background = element_blank())

h.train <- train
h.train$STARS[is.na(h.train$STARS)] <- 0
cbPalette <- c("#58BFFF", "#3300FF", "#E69F00", "#009E73", "#CC79A7")
hist.STARS <- h.train[, c('TARGET', 'STARS')] %>%
  gather(-STARS, key = "var", value = "val") %>%
  ggplot(aes(x = val, fill=factor(STARS))) +
  geom_bar(alpha=0.8) +
  facet_wrap(~ STARS, scales = "free") +
  scale_fill_manual("STARS", values = cbPalette) +
  xlab("TARGET") +
  ylab("") +
  theme(panel.background = element_blank(), legend.position="top")
hist.STARS

hist.LabelAppeal <- h.train[, c('TARGET', 'LabelAppeal')] %>%
  gather(-LabelAppeal, key = "var", value = "val") %>%
  ggplot(aes(x = val, fill=factor(LabelAppeal))) +
  geom_bar(alpha=0.8) +

```

```

facet_wrap(~ LabelAppeal, scales = "free") +
scale_fill_manual("LabelAppeal", values = cbPalette) +
xlab("TARGET") +
ylab("") +
theme(panel.background = element_blank(), legend.position="top")
hist.LabelAppeal
#c("#58BFFF", "#3300FF")

# Boxplot

scaled.train.num <- as.data.table(scale(train[, c('AcidIndex','Alcohol', 'Density',
'Sulphates', 'pH', 'TotalSulfurDioxide','FreeSulfurDioxide',
'ResidualSugar', 'CitricAcid', 'VolatileAcidity','FixedAcidity')], na.rm=TRUE))

melt.train <- melt(scaled.train.num)

scaled_boxplots <- ggplot(melt.train, aes(variable, value)) +
geom_boxplot(width=.5, fill="#58BFFF", outlier.colour="red", outlier.size = 1) +
stat_summary(aes(colour="mean"), fun.y=mean, geom="point",
size=2, show.legend=TRUE) +
stat_summary(aes(colour="median"), fun.y=median, geom="point",
size=2, show.legend=TRUE) +
coord_flip() +
#scale_y_continuous(labels = scales::comma,
#                     breaks = seq(0, 110, by = 10)) +
labs(colour="Statistics", x="", y "") +
scale_colour_manual(values=c("#9900FF", "#3300FF")) +
theme(panel.background=element_blank(), legend.position="top")
scaled_boxplots

# Scatter plot between numeric predictors and the TARGET
linearity <- train %>%
gather(-TARGET, key = "var", value = "value") %>%
ggplot(aes(x = value, y = TARGET)) +
geom_point(alpha=0.1) +
stat_smooth() +
facet_wrap(~ var, scales = "free", ncol=3) +
ylab("TARGET") +
xlab("") +
theme(panel.background = element_blank())

# Scatter plot between log transformed predictors and the log transformed TARGET
# filtered for rows where TARGET is greater than 0
logged_vals <- train[,c('AcidIndex','Alcohol', 'Density',
'Sulphates', 'pH', 'TotalSulfurDioxide','FreeSulfurDioxide',
'Chlorides', 'ResidualSugar', 'CitricAcid', 'VolatileAcidity',
'FixedAcidity', 'TARGET')]
logged_vals <- logged_vals %>%
filter(TARGET>0) %>%
log()

linearity_log <- logged_vals %>%
gather(-TARGET, key = "var", value = "value") %>%

```



```

train_plusmin <- train_imputed_raw

# list of columns that will be transformed
cols <- c("FixedAcidity", "VolatileAcidity",
         "CitricAcid", "ResidualSugar",
         "Chlorides", "FreeSulfurDioxide",
         "TotalSulfurDioxide", "Sulphates", "Alcohol")

# Transformation of train_plusmin by adding the minimum value plus one
for (col in cols) {
  train_plusmin[, col] <- train_plusmin[, col] + abs(min(train_plusmin[, col])) + 1
}

train_plusmin$STARS <- as.factor(train_plusmin$STARS)
train_plusmin$LabelAppeal <- as.factor(train_plusmin$LabelAppeal)
#-----
##3. Data by Jeremy's method
# arithmetically scaled from lower bound of IQR*1.5 to 0, and lesser values
# dropped: train_minscaled
# Subset variables with values for frequencies / concentrations / amounts
# that are < 0
train_scaling_subset <- train_imputed_raw %>%
  dplyr::select(FixedAcidity,
                VolatileAcidity,
                CitricAcid,
                ResidualSugar,
                Chlorides,
                FreeSulfurDioxide,
                TotalSulfurDioxide,
                Sulphates)
# dplyr::rename_all(paste0, '_scaled')

# Function to additively scale values by amount equivalent to lower bound of 1.5 * IQR
# then drop anything below 0 and leaves NAs as they are
positive_scale <- function(x) {
  low_bound <- mean(x, na.rm = TRUE) - (stats:::IQR(x, na.rm = TRUE) * .5) * 1.5
  if(is.na(x)) {
    x = NA
  } else if(x < low_bound) {
    x = 0
  } else {
    x = x + abs(low_bound)
  }
}

# Rescale subset of variables with values < 0
train_iqrscaled_subset <- lapply(train_scaling_subset,
                                    FUN = function(x) sapply(x, FUN = positive_scale)) %>%
  as.data.frame()

# Join scaled subset back to other variables
train_plusiqr15 <- train_imputed_raw %>%
  dplyr::select(TARGET,

```

```

Density,
pH,
Alcohol,
LabelAppeal,
AcidIndex,
STARS) %>%
cbind(train_iqrscaled_subset)

# Rescale discrete label appeal variable and factorize
train_plusiqr15$LabelAppeal <- train_imputed_raw %>%
  select(LabelAppeal) %>%
  sapply(FUN = function(x) x + 2) %>%
  as.factor()

train_plusiqr15$STARS <- as.factor(train_plusiqr15$STARS)
#-----
##4. Data by ABS and Log

# Convert subset of variables to absolute value
train_scaling_subset2 <- train_imputed_raw %>%
  dplyr::select(FixedAcidity,
                VolatileAcidity,
                CitricAcid,
                ResidualSugar,
                Chlorides,
                FreeSulfurDioxide,
                TotalSulfurDioxide,
                Sulphates,
                Alcohol)

train_absscaled_subset <- lapply(train_scaling_subset2,
                                    FUN = function(x) sapply(x, FUN = abs)) %>%
  as.data.frame()

# lapply(train_absscaled_subset, min)

# Join absolute value-scaled subset back to other continuous variables
train_abs <- train_imputed_raw %>%
  dplyr::select(Density,
                pH,
                AcidIndex) %>%
  cbind(train_absscaled_subset)

# Log-scale all continuous variables, adding constant of 1
train_abslog <- lapply(train_abs, FUN = function(x)
  sapply(x, FUN = function(x) log(x+1))) %>%
  as.data.frame()

# Rescale discrete label appeal variable and factorize
train_abslog$LabelAppeal <- train_imputed_raw %>%
  select(LabelAppeal) %>%
  sapply(function(x) x + 2) %>%
  as.factor()

```



```

colnames(data.prep.comparison) <- prepnames
data.prep.comparison$range <- apply(data.prep.comparison, 1, max) - apply(data.prep.comparison, 1, min)

data.prep.AICs <- AIC(poiss.mod.2,poiss.mod.1,poiss.mod.3,poiss.mod.4)
rownames(data.prep.AICs) <- prepnames
data.prep.AICs

##### THERE WAS NO REAL DIFFERENCE IN THE MODEL FOR THE 4 DIFFERENT DATA #####
##### PREPARATIONS SO I WENT WITH THE ONE THAT MADE MOST SENSE TO ME #####
#####

poiss.mod.null = glm(TARGET ~ 1, family = "poisson", data = train_plusmin)

#-----
# REFINEMENT >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

# forward.poiss.mod.1 <- step(poiss.mod.1, direction = "forward", trace=FALSE)
# backward.poiss.mod.1 <- step(poiss.mod.1, direction = "backward", trace=FALSE)

step.poiss.mod.1 <- step(poiss.mod.null, scope = list(upper=poiss.mod.1),
    direction = "both", data = data, trace=FALSE)

drop1.poiss.mod.1 <- drop1(step.poiss.mod.1, test="F")
drop1.poiss.mod.1

# got rid of Ph from step model
poiss.mod.2 <- glm(TARGET ~ STARS + LabelAppeal + AcidIndex + VolatileAcidity +
    TotalSulfurDioxide + Alcohol + FreeSulfurDioxide + Sulphates +
    Chlorides, fam = poisson, d = train_plusmin)
summary(poiss.mod.2)

anova(poiss.mod.2, test="Chisq")

poiss2.influenceplot <- car::influencePlot(poiss.mod.2)

minusinfluential <- train_plusmin[-c(3953, 4940, 8887, 10108, 12513),]

poiss.mod.3 <- glm(TARGET ~ STARS + LabelAppeal + AcidIndex + VolatileAcidity +
    TotalSulfurDioxide + Alcohol + FreeSulfurDioxide + Sulphates +
    Chlorides, fam = poisson, d = minusinfluential)

summary(poiss.mod.3)

anova(poiss.mod.3, test="Chisq")

#-----
# Poisson and Quasipoisson comparison >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
quasi.mod.null <- glm(TARGET~1, fam = quasipoisson, d = minusinfluential)
summary(quasi.mod.null)
```

```
quasi.mod.1 <- glm(TARGET~, fam = quasipoisson, d = minusinfluential)
summary(quasi.mod.1)
```







```
pois.quasi.compare

# Doesn't work for hurdle or zero inflated models
pois_plot <- autoplot(pois.mod.3, which = 1:6, colour = "#58BFFF",
                      smooth.colour = 'red', smooth.linetype = 'solid',
                      ad.colour = 'black',
                      label.size = 3, label.n = 5, label.colour = "#3300FF",
                      ncol = 2) +
  theme(panel.background=element_blank())

quasi_plot <- autoplot(quasi.mod.2, which = 1:6, colour = "#58BFFF",
                      smooth.colour = 'red', smooth.linetype = 'solid',
                      ad.colour = 'black',
                      label.size = 3, label.n = 5, label.colour = "#3300FF",
                      ncol = 2) +
  theme(panel.background=element_blank())
quasi_plot

#### Poisson Model Predictions
pois.pred.raw <- predict(pois.mod.2, newdata = train_plusmin)

#-----
##### NEGATIVE BINOMIAL MODELS >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

# BUILD MODELS
# ZERO-INFLATED NEGATIVE BINOMIAL MODEL

nb_vars <- c('TARGET',
            'FixedAcidity',
            'VolatileAcidity',
            'CitricAcid',
            'ResidualSugar',
            'Chlorides',
            'FreeSulfurDioxide',
            'TotalSulfurDioxide',
            'Density',
            'pH',
            'Sulphates',
            'Alcohol',
            'LabelAppeal',
            'AcidIndex'
      )
)

zinb_vars <- c('TARGET',
               'Chlorides',
               'Density',
               'pH',
               'Sulphates',
               'LabelAppeal',
               'AcidIndex',
               'STARS',
               colnames(train_scaling_subset)
)
)
```

```

# 1st approach with data = train_imputed
#nb.model.jeremy.1 <- glm.nb(formula = TARGET ~ .,
#                             data = dplyr::select(train_selected, nb_vars))
#summary(nbmj1 <- MASS::stepAIC(nb.model.jeremy.1))
# AIC = 51493

# 2nd approach with data = train_plus,min
nb.model.jeremy.2 <- glm.nb(formula = TARGET ~ .,
                             data = train_plusmin)
summary(nbmj2 <- MASS::stepAIC(nb.model.jeremy.2))
# AIC = 45620
# Alcohol, LabelAppeal, AcidIndex, STARS, Volatile Acidity, Chlorides, FreeS02,
# TotalS02, Sulphates all stat sig
# Coefficients make intuitive sense, with sales drivers like more STARS,
# followed by LabelAppeal, then more acidity

# 3rd approach with data = train_plusiqr15
nb.model.jeremy.3 <- glm.nb(formula = TARGET ~ .,
                             data = train_plusiqr15)
summary(nbmj3 <- MASS::stepAIC(nb.model.jeremy.3))
# AIC = 45620
# Same covariates, same coefficients

# 4th approach with data = train_abslog
nb.model.jeremy.4 <- glm.nb(formula = TARGET ~ .,
                             data = train_abslog)
summary(nbmj4 <- MASS::stepAIC(nb.model.jeremy.4))
# AIC = 46319

# Model 1:

neg.bin.imputed <- glm.nb(TARGET ~ ., data=train_imputed)
neg.bin.min <- glm.nb(TARGET ~ ., data=train_plusmin)
neg.min.iqr <- glm.nb(TARGET ~ ., data=train_plusiqr15)
neg.min.abslog <- glm.nb(TARGET ~ ., data=train_abslog)

# Model 3: Zero Dispersion Counts

zero.infl.imputed <- zeroinfl(TARGET ~ . | STARS, data=train_imputed, dist="negbin")
#zero.infl.min <- zeroinfl(TARGET ~ . | STARS, data=train_plusmin, dist="negbin")
#zero.infl.iqr <- zeroinfl(TARGET ~ . | STARS, data=train_plusiqr15, dist="negbin")
#zero.infl.abslog <- zeroinfl(TARGET ~ . | STARS, data=train_abslog, dist="negbin")

#-----
##### MULTIPLE LINEAR REGRESSION MODELS >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
# ----- MODELS -----
# One of two Multiple Linear Regression models.
# Multiple R2 = 0.537. Adjusted R2 = 0.537. P-value = <0.0000000000000002.
# Mean of Residuals is 0. Sum of Residuals is 0.0000000000000079.

```

```

# BIC of 43315
mlr_1 = lm(TARGET ~ AcidIndex + LabelAppeal + STARS, data=train_imputed)
# using train_plusmin has same results.
# using train_abslog has similar, lesser (0.536) results.
# using train_plusiqr15 has same results.
mlr_1_AIC = AIC(mlr_1)
mlr_1_BIC = BIC(mlr_1)

# Two of two Multiple Linear Regression models.
# Multiple R2 = 0.567. Adjusted R2 = 0.566. P-value = <0.0000000000000002.
# Mean of Residuals is -0.0000000000000049. Sum of Residuals is -0.000000000000062.
# BIC of 42650
mlr_2 = lm(TARGET ~ AcidIndex*LabelAppeal*STARS, data=train_imputed)
# using train_plusmin has same results.
# using train_abslog has similar, lesser (0.567 and 0.565) results.
# using train_plusiqr15 has same results.
mlr_2_AIC = AIC(mlr_2)
mlr_2_BIC = BIC(mlr_2)

# ----- PREDICTIONS -----

mlr_1_predictions = predict(mlr_1, data=train_imputed)
mlr_1_predictions = as.factor(round(mlr_1_predictions))
levels(mlr_1_predictions) = sort(as.numeric(unique(c(-2, train_imputed$TARGET, levels(mlr_1_predictions)
mlr_1_confusion = confusionMatrix(mlr_1_predictions, reference=factor(train_imputed$TARGET, levels=leve

mlr_2_predictions = predict(mlr_2, data=train_imputed)
mlr_2_predictions = as.factor(round(mlr_2_predictions))
levels(mlr_2_predictions) = levels(mlr_1_predictions)
mlr_2_confusion = confusionMatrix(mlr_2_predictions, reference=factor(train_imputed$TARGET, levels=leve

# ----- PLOTS -----

# mlr_1 plot. Q-Q looks good.
mlr_1_plot = autoplot(mlr_1, which = 1:6, colour = "#58BFFF",
                      smooth.colour = 'red', smooth.linetype = 'solid',
                      ad.colour = 'black',
                      label.size = 3, label.n = 5, label.colour = "#3300FF",
                      ncol = 2) +
  theme(panel.background=element_blank())

# mlr_2 plot. Q-Q doesn't look as good, but Residuals vs Leverage looks better.
mlr_2_plot = autoplot(mlr_2, which = 1:6, colour = "#58BFFF",
                      smooth.colour = 'red', smooth.linetype = 'solid',
                      ad.colour = 'black',
                      label.size = 3, label.n = 5, label.colour = "#3300FF",
                      ncol = 2) +
  theme(panel.background=element_blank())

#-----
##### MODEL SELECTION >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```

model_select_lt <- rbind(c('Poisson-Logit Hurdle Regression',useBIC(hurd.mod2, return.K = FALSE, nobs =
c('Zero-inflated Poisson',useBIC(zi.mod2, return.K = FALSE, nobs = NULL),logLik(zi.mod2),
c('Negative Binomial 1',useBIC(nb.model.jeremy.2, return.K = FALSE, nobs = NULL),
c('Negative Binomial 2',useBIC(nb.model.jeremy.3, return.K = FALSE, nobs = NULL),
c('Linear Model',useBIC(mlr_1, return.K = FALSE, nobs = NULL),logLik(mlr_1),'1',
c('Linear Model 2',useBIC(mlr_2, return.K = FALSE, nobs = NULL),logLik(mlr_2),

)

colnames(model_select_lt) <- c('model','BIC','Log-likelihood', 'Degrees of freedom')

root.pois <- countreg::rootogram(hurd.mod2, style = "hanging", plot = FALSE)
ylims <- ylim(-1, 100)
countreg_model <- plot_grid(autoplot(root.pois) + ylims, ncol = 1, labels = "auto")

test$TARGET_hurd <- predict(hurd.mod2, newdata = test, type = "response")

predictions_lt <- round(test$TARGET_hurd)

p1.pred <- ggplot(data.frame(predictions_lt), aes(predictions_lt)) +
  geom_histogram(fill = "#58BFFF", bins = 20) +
  xlab("Test Data Predictions") +
  ylab("") +
  theme(panel.background = element_blank())

p2.pred <- ggplot(train, aes(TARGET)) +
  geom_histogram(fill = "#58BFFF", bins = 20) +
  xlab("Training Data") +
  ylab("") +
  theme(panel.background = element_blank())

summary.pred.count <- describe(test[, c('TARGET_hurd')])[,c(2,8,3,5,9,4)]

```

	Est.	S.E.	t val.	p
(Intercept)	1.98	0.53	3.70	0.00
AcidIndex	-0.12	0.07	-1.76	0.08
LabelAppeal-1	0.82	0.58	1.41	0.16
LabelAppeal0	1.31	0.56	2.33	0.02
LabelAppeal1	1.42	0.58	2.45	0.01
LabelAppeal2	1.41	0.79	1.78	0.08
STARS1	0.70	0.87	0.81	0.42
STARS2	0.66	1.19	0.55	0.58
STARS3	1.16	1.76	0.66	0.51
STARS4	4.27	1.25	3.43	0.00
AcidIndex:LabelAppeal-1	-0.08	0.07	-1.06	0.29
AcidIndex:LabelAppeal0	-0.13	0.07	-1.85	0.06
AcidIndex:LabelAppeal1	-0.17	0.07	-2.30	0.02
AcidIndex:LabelAppeal2	-0.14	0.10	-1.49	0.14
AcidIndex:STARS1	-0.01	0.11	-0.12	0.90
AcidIndex:STARS2	0.07	0.15	0.43	0.67
AcidIndex:STARS3	0.06	0.23	0.27	0.79
AcidIndex:STARS4	0.09	0.16	0.59	0.56
LabelAppeal-1:STARS1	0.65	0.93	0.69	0.49
LabelAppeal0:STARS1	1.55	0.91	1.71	0.09
LabelAppeal1:STARS1	1.50	0.96	1.56	0.12
LabelAppeal2:STARS1	-2.67	1.51	-1.77	0.08
LabelAppeal-1:STARS2	0.28	1.25	0.22	0.82
LabelAppeal0:STARS2	0.68	1.22	0.56	0.58
LabelAppeal1:STARS2	1.45	1.24	1.17	0.24
LabelAppeal2:STARS2	1.52	1.57	0.97	0.33
LabelAppeal-1:STARS3	-0.49	1.89	-0.26	0.80
LabelAppeal0:STARS3	0.13	1.79	0.07	0.94
LabelAppeal1:STARS3	0.77	1.81	0.43	0.67
LabelAppeal2:STARS3	1.78	1.97	0.90	0.37
LabelAppeal-1:STARS4	-3.34	1.93	-1.73	0.08
LabelAppeal0:STARS4	-1.95	1.42	-1.37	0.17
LabelAppeal1:STARS4	-2.11	1.36	-1.55	0.12
LabelAppeal2:STARS4	NA	NA	NA	NA
AcidIndex:LabelAppeal-1:STARS1	-0.03	0.12	-0.26	0.79
AcidIndex:LabelAppeal0:STARS1	-0.09	0.12	-0.77	0.44
AcidIndex:LabelAppeal1:STARS1	-0.03	0.12	-0.22	0.82
AcidIndex:LabelAppeal2:STARS1	0.40	0.18	2.18	0.03
AcidIndex:LabelAppeal-1:STARS2	0.02	0.16	0.15	0.88
AcidIndex:LabelAppeal0:STARS2	0.07	0.16	0.48	0.63
AcidIndex:LabelAppeal1:STARS2	0.10	0.16	0.62	0.53
AcidIndex:LabelAppeal2:STARS2	0.17	0.20	0.85	0.40
AcidIndex:LabelAppeal-1:STARS3	0.14	0.25	0.57	0.57
AcidIndex:LabelAppeal0:STARS3	0.15	0.23	0.62	0.53
AcidIndex:LabelAppeal1:STARS3	0.18	0.24	0.78	0.44
AcidIndex:LabelAppeal2:STARS3	0.15	0.25	0.58	0.56
AcidIndex:LabelAppeal-1:STARS4	0.20	0.25	0.80	0.43
AcidIndex:LabelAppeal0:STARS4	0.06	0.18	0.34	0.73
AcidIndex:LabelAppeal1:STARS4	0.19	0.17	1.12	0.26
AcidIndex:LabelAppeal2:STARS4	NA	NA	NA	NA

Table 6: Comparison of models

model	BIC	Log-likelihood	Degrees of freedom
Poisson-Logit Hurdle Regression	40816.6060825119	-20280.6413861792	27
Zero-inflated Poisson	40914.6863017303	-20329.6814957884	27
Negative Binomial 1	45786.5052909723	-22789.2277382516	23
Negative Binomial 2	45775.2523193232	-22783.6012524271	22
Linear Model	43336.4446687359	-21616.2098807507	11
Linear Model 2	42836.2580131354	-21186.4371677271	49

Table 7: TARGET Predictions

	n	min	mean	median	max	sd
X1	2523	0.11	3	3	6.8	1.4